ANO 1 - NÚMERO 10
Cr$ 700,00

interface

Hardware - Software - Computadores - Mini - Micro - Teleprocessamento

CURSO DE BASIC
(1º LIÇÃO)

PASCAL

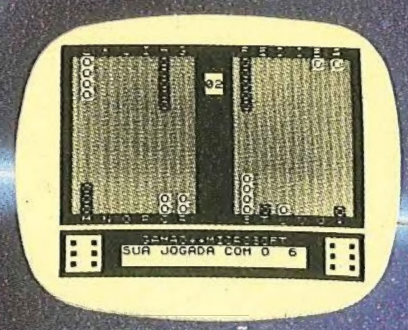
SIMULADOR DE VÔO

SUPLEMENTO DO PEQUENO USUÁRIO

CONCURSO
TK 82-C

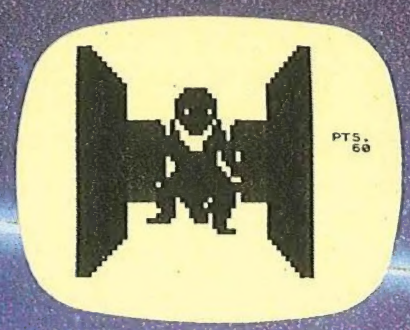
MICROSOFT

Programas para o seu TK82-C e TK85



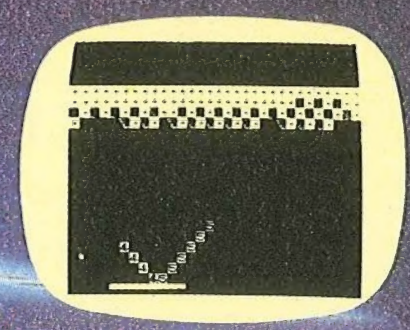
JOGO DE GAMÃO
16K

Este programa apresenta o tabuleiro no vídeo e utiliza o eficiente código de máquina, permitindo 4 (quatro) níveis de dificuldades de jogo.



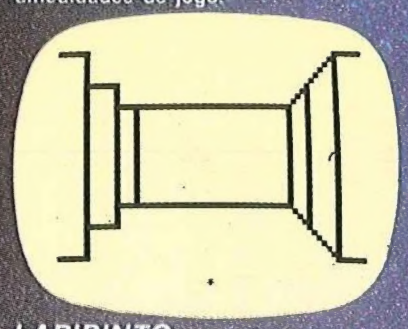
MONSTRO DAS TREVAS TRIDIMENSIONAL · 16K

Impressionante jogo onde você deve evitar o monstro das trevas. Tudo em 3 dimensões.



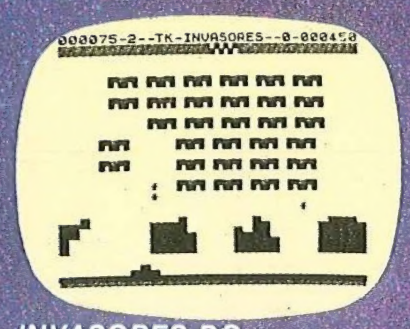
DEMOLIDOR
2K

Jogo animado, tipo "fliperama". O jogador deverá demolir uma parede com uma bola que se encontra sempre em movimento.



LABIRINTO TRIDIMENSIONAL · 16K

Jogo em três dimensões. O jogador poderá definir a dificuldade do Labirinto. O programa apresenta a posição do jogador em perspectiva. Em qualquer momento é possível pedir auxílio ao computador.

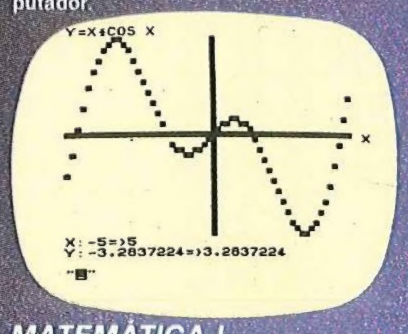


INVASORES DO ESPAÇO · 16K

Consiste de uma frota de naves invasoras extraterrenas, descendo no planeta Terra. Sua missão é destruir as naves invasoras dispondo da arma de raios laser.

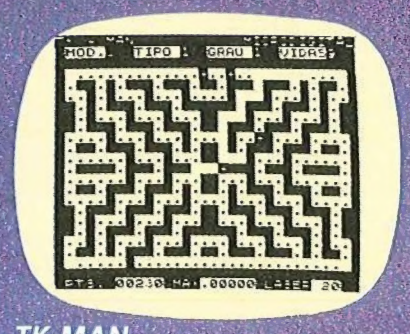
RALLY
16K

Emocionante corrida de rally em um labirinto, onde poderão ser testados sua habilidade e seus reflexos. Para conseguir seu intento, você deverá evitar carros-ataque e obstáculos em seu trajeto.



MATEMÁTICA I
16K/64K

Análise gráfica de funções matemáticas, resolução de sistemas de equações lineares (16K-51 equações/64K-95 equações), e Cálculo de integrais definidos.



TK-MAN
16K

Jogo animado onde deverão ser apagados todos os pontinhos espalhados em um labirinto (o programa contém 15 tipos de labirinto). Você será impedido a qualquer custo, por 4 extraterrenos, guardiães do labirinto, que poderão ser combatidos com cargas de raios-laser.

T-KALC
16K/64K

Programa desenvolvido para cálculos numéricos em planilha. O usuário define as colunas, as linhas e as fórmulas aplicadas. Similar ao famoso Visicalc. De grande versatilidade, este programa permite a formulação de cálculos científicos e comerciais, análise de tabelas numéricas e outras aplicações.

TKADREZ II
16K

Este jogo apresenta o tabuleiro e as peças no vídeo. Permite a escolha de até 7 níveis de dificuldade. O programa fornece a qualquer momento, a listagem dos lances efetuados, e armazena em fita a posição das peças. Ele poderá recomendar a sua jogada.

NOS REVENDEDORES AUTORIZADOS EM TODO PAÍS

80 programas à sua disposição. Solicite folheto.

MICROSOFT®

Rua do Bosque, 1.234 - PABX 825-3355
CEP 01136 - Cx. Postal 54.121 - São Paulo-SP

conteúdo

Interface

matérias

Pascal: Uma Opção para o Programador Sério 06
Problemas com o BASIC, programação estruturada e aplicações.
Por Claudio Esperança

Filia: Um Sistema de Programação para Microcomputadores (3ª Parte) 11
O interpretador interno e a organização do dicionário em FILIA.
Por Rildo Pragana

Curso Microprocessador Z-80 (9ª Lição) 18
Micro-Poster com o conjunto de instruções do Z-80
Por André Gil Rubens e Ney Acyr de Oliveira (A & N – Consultoria em Hardware e Software de Microcomputadores)

Memórias (3ª Parte) 41
Memórias semicondutoras e expansão de memória.
Por Cesar da Costa

Mumps: Características e Comparações com Outros Sistemas de Programação (3ª Parte) 48
Sistema de Banco de Dados, gerência de registros e estrutura de nodo.
Por T. Munneke; R. F. Walters; J. Bowie; C. B. Lazarus e D. A. Bridger (Tradução MEDIDATA - RJ).

Mercado de Computadores

departamentos

Notas do Editor 02
Novos Produtos 03
Informe 04
Consultoria 10
Calendário 22
Cartas 44
Livros 46

seções

Educação / Tecnologia 40
A Automação Bancária
Por Juarêz Fernandes Ramos

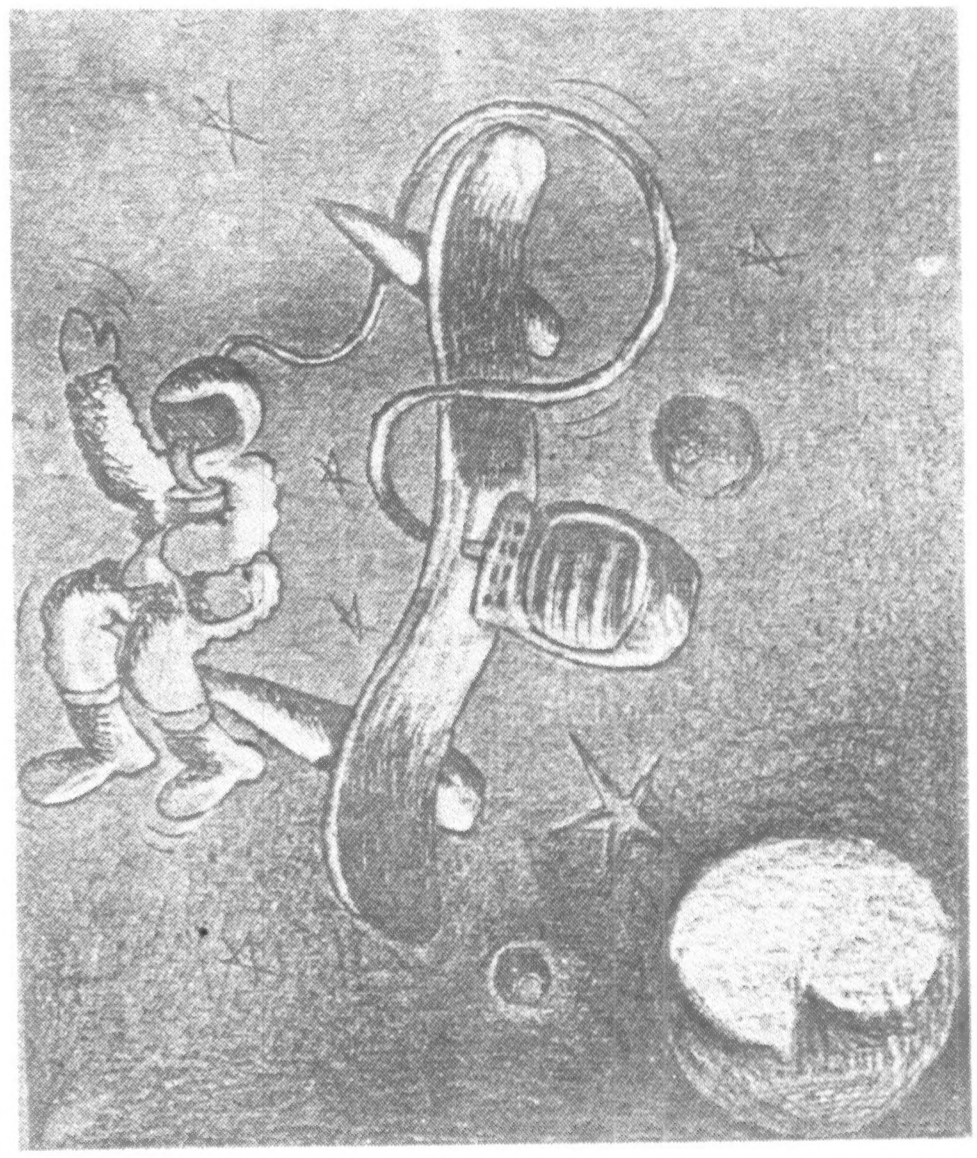
SUPLEMENTO DO PEQUENO USUÁRIO DE MICROCOMPUTADORES

Curso de Programação Basic (1ª Lição) 23
Constituição do computador, algoritmos e exercícios
Por José Arthur da Rocha

Programa: Simulador de Vôo 27
Decodificação da linha REM e listagem

Microcomputadores 37
Arquivo de Dados em Disco versus Arquivo de Dados em Cassete
Por Cesar da Costa

Glossário de Termos Técnicos 38
As rápidas mudanças dentro do campo da Informática levam a um vocabulário que rapidamente muda. Além disso, poucas pessoas dominam a língua inglesa. Palavras de **A** a **Z** em 12 edições.



SIMULADOR DE VÔO 27

DIREÇÃO:
Naila Glória O. Côrtes
Cesar da Costa

EDITORIA:
José Levy Fidelix da Cruz

EDITORIA TÉCNICA:
Naila Glória O. Côrtes
Cesar da Costa

CONSULTORIA TÉCNICA:
Ivan Costa
Tarcísio Neves da Cunha
Manuel Lois Anido
José Arthur da Rocha
André Gil Rubens
Ney Acyr de Oliveira
Cesar de Araújo Lima

PRODUÇÃO/DIAGRAMAÇÃO:
Nilton Luiz Côrtes

ARTE FINALIZAÇÃO:
Jorge Gatto
Sandra Maria Pinto dos Santos

REDAÇÃO/REVISÃO:
Nelma Bernadete C. de Castro

DIREÇÃO COMERCIAL/PUBLICIDADE:
SUCESS – Assessoria, Planejamento e Marketing Ltda
Diretor: José Levy Fidelix da Cruz

Publicidade/RJ:
Rua Senador Dantas, 117 sala 1139
Tel.: (021) 240-5200
José Augusto Nunes Rodrigues
Roberto Passari

Publicidade/SP:
Av. Paulista, 1139 – Cj. 801
Tel.: (011) 284-8384
Araci Monteiro Pedreira

Publicidade/RS:
Rua Osório Ramos Correa, 220
Tel.: (0512) 88-1252
Francisco Link

ASSINATURAS:
Gerente Nacional:
José Augusto R. Pereira

Supervisor/RJ:
Edson Teixeira Pedroso

Supervisor/SP:
Luiz Henrique C. Guimarães

Representante/DF:
Compushow Computadores Ltda
Tel.: (061) 273-2128

Representante/BA:
Oficcina Minis e Microcomputadores Ltda
Tel.: (071) 248-6666 R/268
Preços:
Assinaturas/Brasil – Cr$ 7.500,00 (1 ano)
Cr$ 15.000,00 (2 anos)
Preços especiais para vendedores/PRODIGT em outros estados – Cr$ 15.000,00 (1 ano)
Cr$ 20.000,00 (2 anos)
Assinaturas/Exterior – US$ 50,00 (1 ano)
US$ 100,00 (2 anos)

TIRAGEM:
30.000 exemplares

DISTRIBUIÇÃO:
Abril S/A. Cultural e Industrial – Divisão Distribuição (exclusivo para todo o Brasil)

COMPOSIÇÃO:
Damião Moraes

FOTOLITOS/IMPRESSÃO:
Gráfica Barbero

Interface é editada pela Prodigt – Processamento, Tecnologia e Comunicação Ltda.
Administração e Redação:
Estrada do Tindiba, 2380 – Jacarepaguá – Rio de Janeiro – RJ – CEP 22700 – Tel.: 392-8965.
Interface não aceita matéria redacional paga.
Todos os direitos de reprodução desta publicação estão reservados, só podendo ser reproduzidos com fins comerciais, mediante autorização prévia.
Os artigos assinados são de responsabilidade do autor, as opiniões emitidas não são necessariamente coincidentes com as da Revista.
Registro no INPI protocolo nº 810965755.

NOTAS DO EDITOR

"É extremamente difícil profetizar – especialmente com relação ao futuro" (provérbio chinês).

Várias questões se levantam e com a mesma urgência clamam por uma definição. Por mais reflexão que se invista e análise que se utilize na situação como um todo, transparece a dúvida se, por outro lado, prenunciam a reserva dos que se atrevem a fazer algum tipo de previsão. Nós ousamos apostar no talvez acreditando no fenômeno Informática enquanto meio para – "alcançarmos patamares mais altos de qualidade de vida" (Hélio Azevedo).

A nível de política nacional existem implicações que vêm se tornando tema constante nas páginas dos jornais. Mas, nossa esperança não vai acabar; a realidade já está mais perto. Até o final do ano, o Congresso deve votar o projeto de lei que institucionaliza a política nacional de informática e dá incentivos para as áreas de microeletrônica, automação, instrumentação e processamento de dados.

A Automação Industrial foi debatida recentemente durante o 1º Congresso Nacional de Automação Industrial – CONAI, onde se buscou através da discussão os verdadeiros rumos que poderá assumir este processo no Brasil.

Com características de irreversibilidade, este processo deverá ser conduzido (E bem), no entanto, de maneira que as conseqüências no âmbito sócio-econômico, como por exemplo o desemprego, sejam minimizadas. Segundo a SEI – Secretaria Especial de Informática, o Congresso vai legislar também sobre a automação, estando esta incorporada ao projeto que será votado e que foi elaborado inicialmente pela deputada Cristina Tavares.

A área de Comandos Numéricos tem três anos para absorver o know-how importado, e a indústria brasileira de Comandos Numéricos Computadorizados (CNC's) deve estar capacitada para tal. Esses equipamentos construídos com base em microprocessadores substituem com vantagem a mão-de-obra industrial. Esta é a conseqüência temível, a que justifica mais e mais a condução do processo de maneira firme e segura.

As indefinições, no entanto, não têm conseguido impedir o crescimento do setor da informática nacional, que cresce 50% ao ano; índice que corrobora a suspeita de que vamos chegar com segurança.

Devido ao grande número de solicitações no sentido da publicação de um suplemento dedicado aos usuários de pequenos sistemas de microcomputadores, após longo período de elaboração, nesta edição iniciamos o SUPLEMENTO DO PEQUENO USUÁRIO DE MICROCOMPUTADOR. Neste, trazemos o **Curso de Programação Basic,** *em doze lições, preparado especialmente para os iniciantes; o* **Glossário de Termos Técnicos,** *da letra* **A** *a* **Z,** *apresentando os principais termos técnicos absorvidos do inglês e utilizados em computação. Também no SUPLEMENTO programas para aplicação imediata, trazendo nesta edição o* **Simulador de Vôo,** *com a decodificação da primeira linha REM. Aguardem também muitas dicas de hardware e software dos micros TK82, TK85, CP200, DGT-100, CP-500 e muitos outros.*

O software, nesta edição, continua trazendo para os programadores e analistas: FILIA: Um Sistema de Programação Para Microcomputadores, MUMPS: Características e Comparações Com Outros Sistemas de Programação e PASCAL: Uma Opção Para o Programador Sério.

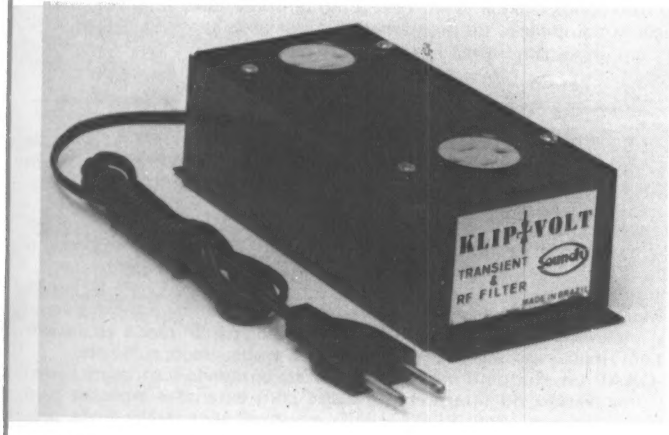
A tônica do hardware nesta edição vem com MEMÓRIAS (Semicondutoras) e o CONJUNTO DE INSTRUÇÕES DO MICROPROCESSADOR Z-80. E no próximo número, CHIPS (Geradores e Testadores de Paridade) e PERIFÉRICOS (Barramento S-100).

Desta forma julgamos estar ampliando ainda mais nossos canais de comunicação com os leitores.

NOVOS PRODUTOS

KLIP VOLT – SUPRESSOR DE TRANSIENTES E RF PARA MICROCOMPUTADORES

A Soundy-Indústria Eletrônica Ltda, após pesquisas que visavam determinar a causa das avarias em aparelhos eletrônicos (microcomputadores, periféricos, etc.) cujos circuitos eram perfeitamente dimensionados e, portanto, deveriam funcionar sem problemas por muitos anos, constatou que na maioria das vezes as "panes" eram provocadas por transientes existentes na rede domiciliar de energia elétrica. Esta ao ser monitorada apresentava picos de até 5000 volts que, embora de curtíssima duração, eram suficientes para danificar semicondutores ou outros componentes. Assim sendo desenvolveu o **Klip Volt**, dispositivo a ser intercalado entre o equipamento e a rede elétrica, baseado em semicondutores especiais MOV, que além de proteger os equipamentos contra os transientes atua como filtro de RF. O **Klip Volt** é produzido para uso em redes de T10/127V e permite controle de até 880W.



DATA MANAGER DA MONK

A Monk Micro Informática, dedicada à confecção de software de qualidade e dentro das necessidades do mercado brasileiro, lança o **Data Manager**, mais conhecido como Gerenciador de Banco de Dados (GBD). Esse programa vem resolver uma série de problemas dos profissionais liberais como médicos, dentistas, advogados e engenheiros, que estão sempre às voltas com agendas, honorários em pendência, remanejamento de consultas, listagem de clientes e andamento de processos.

O **Data Manager** é um pacote de programação que transforma seu micro num sistema de arquivo bem organizado, com várias possibilidades de manipulação dos dados que você arquiva. Apesar de ser basicamente um banco de dados, faz quatro operações numéricas podendo guardar dados como somas de dinheiro e outros valores, e realizando cálculos de pagamentos efetuados.

O **Data Manager** roda nos micro CP500, Naja, TRS 80III ou qualquer outro compatível. Com dois drivers e opcionalmente uma impressora, faz também impressão de etiquetas e relatórios diversos. Este utilíssimo programa pode ser encontrado ao preço médio de Cr$ 100.000,00 em todos os revendedores Monk, juntamente com um termo de garantia Monk e o manual explicativo, que lhe mostrará inúmeras aplicações.

NOVOS JOGOS ODYSSEY

A coleção de jogos para o videogame Odyssey acaba de ser acrescida de mais quatro cartuchos que, com os quinze anteriores, apresenta um total de dezenove cartuchos e vinte e quatro jogos diferentes. Três deles pertencem à série esportiva e são: "Bilhar", "Futebol de Salão"/"Hokey" e "Esqui nos Alpes". O último deles é da série ação e apresenta um "Duelo no Velho Oeste".

Estes são os **novos jogos Odyssey**:

"Bilhar", com dois tipos de jogos: o "Bola Oito" ("Eight Ball"), disputado com duas bolas pretas, o jogador que primeiro encaçapar uma delas ganha a partida; e o "Rotação", onde o jogador deve encaçapar o maior número de bolas para ganhar o jogo. Para dois parceiros.

"Futebol de Salão"/"Hockey", dois jogos em um só cartucho, permite, no primeiro, que os jogadores chutem, cabeceiem e, principalmente, marquem os gols. Já o "Hockey", na versão sobre o gelo, é disputado em três tempos, com um taco para cada jogador. Ambos possuem diversos graus de dificuldades. Para dois parceiros.

"Esqui nos Alpes" compreende três tipos de competições: "Downhill", descida direta em alta velocidade; "Slalom", descida sinuosa demarcada por bandeiras da cor do jogador; e "Giant-slalom", com as mesmas regras do "slalom", porém mais longa.

"Duelo no Velho Oeste" é uma luta entre dois pistoleiros e aquele que atingir dez vezes o seu adversário ganha a partida. Há esconderijos para proteção e remuniciamento. Pode ser jogado em dois parceiros ou sozinho, contra a máquina, aliás uma temível adversária.

INFORMÁTICA RECEBE APOIO DO CLUBE DE ENGENHARIA

O Clube de Engenharia engaja-se em mais uma luta a favor dos profissionais de Engenharia: o setor de informática. Segundo seu presidente, Matheus Schnaider, um dos primeiros passos nesta luta, na busca de evolução do setor, é a institucionalização da Política Nacional de Informática, com poucas concessões às multinacionais.

Uma das formas de manifestar essa posição foi concretizada através de um almoço realizado no dia 14 de julho, no 24º andar do Clube, onde foram reunidas todas as entidades ligadas ao setor tendo as mesmas sido homenageadas e obtido uma grande oportunidade de se manifestar, expondo suas posições ou decisões em relação ao tema.

Com a representação de várias entidades, realizou-se o almoço sobre um clima bastante agradável e informal, não faltando, porém, colocações de alguns representantes dessas entidades, tais como o Presidente da APPD, Sr. Sérgio Rosa, o Sr. José Maria Sobrinho, representante da ASSESPRO, e a Sra. Marília Rosa Duran, também da APPD, todos demonstrando de forma clara a preocupação com a problemática do setor dentro da atual política do nosso país.

SUCESU

Eventos Especiais

Aplicações da informática com laser, sessões especiais e técnicas, painéis de debates e conferências são alguns dos temas dos Eventos Especiais do Informática 83.

Os painéis de debates terão duração de 1/2 a 1 dia, com apresentação de 20 a 40 minutos sobre determinado assunto e posterior debate entre os convidados e a platéia. Os painéis já definidos são: A Informática e a Produção Industrial, Microeletrônica, A Informática e a Educação, O Impacto da Automação na Sociedade, O Desenvolvimento da Tecnologia e Desenvolvimento de Software Nacional.

Crescem os Debates Sobre a Estatização da Informática

A regulamentação da profissão e o sindicalismo, estatização da Informática, avaliação dos modelos industriais da Informática brasileira e o desenvolvimento de uma política nacional de software são alguns dos principais assuntos que já estão sendo levantados, como temário, para o XVI CONGRESSO NACIONAL DE INFORMÁTICA. Juntamente com a III FEIRA INTERNACIONAL DE INFORMÁTICA, esse encontro será realizado no Parque Anhembi de São Paulo entre os dias 17 e 23 de outubro próximos.

Esses debates, com a participação de entidades como a ABEP, ABICOMP, ANDEI, APPD, ASSESPRO, CENADEM, SBC e TELEBRASIL, estarão abertos ao grande público. Informações complementares e inscrições, para os debates e os dois eventos, poderão ser obtidas na Secretaria do Informática-83, à Avenida Paulista, 1159 – 14º andar, cjs. 1404/5, telefone 288-9452.

MICRO ITAUTEC PARA ÓRGÃO DE PESQUISA

O primeiro microcomputador ITAUTEC I-7000 comercializado e entregue foi instalado no IMPA – Instituto de Matemática Pura e Aplicada, do Conselho de Desenvolvimento Científico e Tecnológico – CNPq, do Rio de Janeiro.

Será usado naquele organismo de pesquisa como instrumento auxiliar nas pesquisas em andamento e em ensino.

O IMPA além de se dedicar a pesquisas em Matemática, Estatística, Pesquisa Operacional e Economia Matemática, forma mestres nessas áreas e doutores em Matemática. Nas áreas de Estatística e Pesquisa Operacional, também são desenvolvidos software estatístico e software para modelos de otimização de grande porte, respectivamente.

O IMPA vai usar o microcomputador ITAUTEC como terminal IBM, em processamento distribuído para edição e compilação de programas voltados às suas atividades específicas, e em ensino utilizando a linguagem Basic.

PSI-GRAF

A PSI-PROJETOS E SERVIÇOS EM INFORMÁTICA IND. E COM. LTDA. acaba de lançar no mercado o **PSI-GRAF**, um dispositivo que transforma um CP-500 normal em um poderoso microcomputador com alta resolução gráfica (512 x 192 pontos) e amplia o vídeo de texto para 80 ou 85 colunas.

Dentre as características do novo produto, destaca-se a possibilidade de "misturar" o vídeo normal de 64 colunas por 16 linhas com gráficos de alta resolução. Dessa forma, o usuário pode inclusive dar **SCROLL** no texto, mantendo os gráficos fixos no vídeo.

Outra característica interessante é a ampliação da capacidade do vídeo para 80 ou 85 colunas, ao invés das 64 colunas originais do equipamento. Esta facilidade é totalmente controlada por software, incluindo a configuração dos caracteres, possibilitando ao usuário criar símbolos utilizados em línguas estrangeiras, como japonês, árabe, grego, russo etc.

Acompanha o **PSI-GRAF** um disquete ou uma fita cassete contendo o software básico de operação, incluindo uma versão do interpretador Basic com extensões especiais para processamento de comandos gráficos, como PLOT, LINE, e outros. Acompanha ainda, programas para ativação do modo 80/85 colunas; configuração de caracteres; diversos utilitários; programas para demonstração; exemplos de utilização e um manual de instruções completo.

Atualmente, este produto está sendo instalado apenas em CP-500 com ou sem unidades de disquetes. Em breve, o **PSI-GRAF** também poderá ser instalado em outros micros compatíveis com o TRS-80 modelo III, tais como: DGT-100, D-8000, JR, NAJA, CP-300 e outros.

O endereço da PSI é Rua Barão do Triunfo, 464 – Cj. 31 – Brooklin – SP., Tel.: (011) 531-9902, Telex: (011) 22966.

TRADUÇÕES – O QUE FALTAVA A INFORMÁTICA

Pioneiro no setor, já está disponível para a comunidade de informática, o apoio de traduções/versões especializadas – português/inglês – inglês/português – para manuais e publicações em geral, atendendo a todas as empresas do ramo. O serviço é feito em três fases principais: interpretação imediata do texto, revisão técnica – nesta fase o trabalho é enviado por malote a consultores técnicos, todos profissionais atuantes na área, de onde retorna para a finalização, quando passa pela revisão ortográfica, é redatilografado e apresentado ao cliente. Todo o trabalho é baseado na qualidade/velocidade exigida pelo cliente.

Informações mais detalhadas pelos telefones: (021) 264-6392, 264-7391, 228-2798.

Ou pelo correio, no endereço: Rua Professor Gabizo, 225 – Tijuca – RJ.

DISCOVER TRADUÇÕES.

informe interface

LOGUS COMPUTADOR

A LOGUS torna-se a primeira empresa 100% brasileira a exportar **Know-How** de computadores para os EUA. Os seus diretores, Jalmar Oliveira e Sergio Campos acabaram de regressar de Boston — Centro Nervoso da Tecnologia americana de computadores — onde mostraram, discutiram e provaram a superioridade de sua tecnologia, obtendo como resultado a venda de dois sistemas para a Arthur D. Little — empresa de consultoria encarregada do **CHECK** e para a Epoxy Technology, empresa multinacional com subsidiárias na Inglaterra, França, Alemanha e Japão, onde futuramente serão instalados equipamentos LOGUS interligados.

GDM INFORMÁTICA

A GDM INFORMÁTICA LTDA está comercializando a linha completa da SYSDATA, DIGITUS e MICRODIGITAL. Estão abertas as inscrições para o curso de BASIC, e em breve será lançada uma nova linha de cursos variados, todos na área da micro-informática.

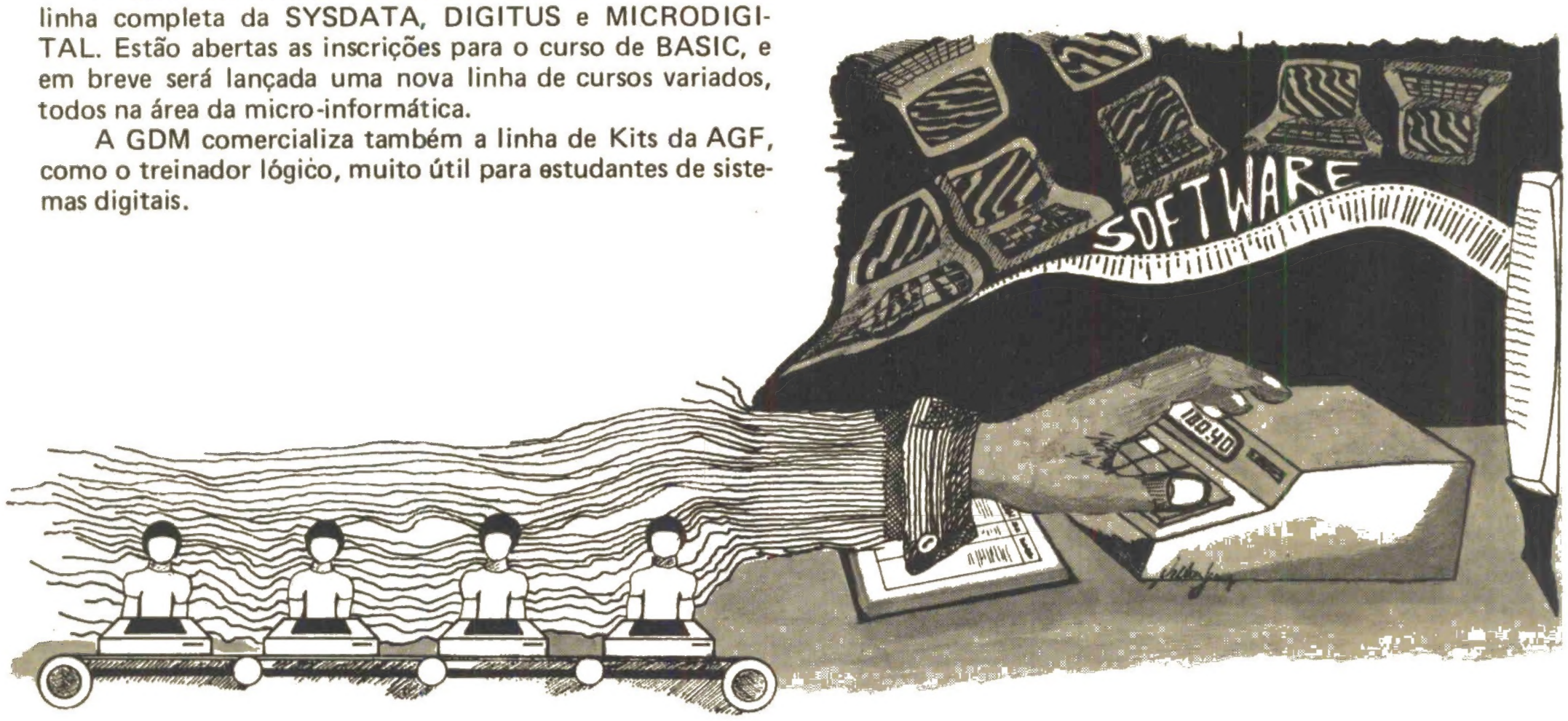
A GDM comercializa também a linha de Kits da AGF, como o treinador lógico, muito útil para estudantes de sistemas digitais.

BRASCOM NA N. C. C. NATIONAL COMPUTER CONFERENCE

Sempre atenta e preocupada em conhecer e adequar sua linha de produtos aos mais recentes lançamentos e inovações de utilidade ao mercado brasileiro, a BRASCOM se fez presente na N. C. C.

Acompanhando a delegação formada por empresários brasileiros e organizada pela ABICOMP, seguiram o Sr. Fernando Calaigian Jorge — DIRETOR PRESIDENTE e seu Gerente de Desenvolvimento.

A constatação do sucesso atual do Sistema UNIX, valoriza ainda mais o Sistema Operacional para o Sistema BR 1000M — Multiusuário da BRASCOM, escrito para operar com o microprocessador Z80A ou B (este com clock de 6Mhz).



A LAURENTI ENTREGA 111 EQUIPAMENTOS PARA A CAIXA ECONÔMICA FEDERAL

Prestigiando o desenvolvimento da tecnologia brasileira, a CAIXA ECONÔMICA FEDERAL adquiriu 111 equipamentos da LAURENTI EQUIPAMENTOS PARA PROCESSAMENTO DE DADOS LTDA.

A entrega de 52 cortadoras eletrônicas M-25 e 59 separadoras SM-3E foi feita no tempo recorde de 50 dias.

O evento vem demonstrar a abertura da CAIXA ECONÔMICA FEDERAL ao valorizar o produto nacional e o desenvolvimento tecnológico da LAURENTI, uma empresa totalmente brasileira, estruturada para produzir equipamentos do mesmo nível dos importados.

Os equipamentos destinam-se aos subcentros da CAIXA ECONÔMICA FEDERAL em vários estados do Brasil.

CONTABILIDADE FACILITADA

Está sendo lançada no mercado a nova versão do **Sistema Cetil de Contabilidade,** que oferece vantagens adicionais aos usuários de software e serviços, por conter aperfeiçoamentos técnicos e operacionais visando agilizar todo o processo de conferência e recuperação das informações.

Entre as inovações, vale citar a possibilidade de emitir o Razão na fase de fechamento, redução e simplificação dos documentos de entrada, número ilimitado de lançamentos por lote, checagem total das informações na fase de fechamento, arquivo único para todas as empresas processadas no CPD e controle automático de salvamento dos arquivos.

O **Sistema Cetil de Contabilidade** já mereceu aprovação de centenas de empresas dos mais diversos ramos de atividade, sendo um dos mais utilizados do país. Entre os usuários figuram Brasimác, Teka, Frigorífico Mouran-Araçatuba, Levi's, Prodesp, Banco Mercantil de São Paulo, Telebahia, Telesc, Banestado e Sadia.

PASCAL

UMA OPÇÃO PARA O PROGRAMADOR SÉRIO

Por Claudio Esperança

QUAL O PROBLEMA COM O BASIC?

Nesses anos de rápida expansão da indústria dos microcomputadores, uma tradição tem se firmado: na grande maioria dos microcomputadores disponíveis no mercado, a principal linguagem de programação é o **BASIC**. A razão dessa preferência quase unânime é fácil de entender: o BASIC é uma linguagem acessível ao iniciante, de fácil aprendizado. Isso se deve principalmente ao fato de que o BASIC é uma linguagem projetada para ser utilizada por principiantes, como o próprio nome diz (BASIC, além de significar "básico" em inglês, também é sigla de Beginner's All-purpose Symbolic Code). Aliado a isso, um fator também importante para a popularidade da linguagem é a natureza essencialmente interativa de sua estrutura. Mais ainda: o BASIC é geralmente implementado através de um interpretador, isto é, o programador pode digitar seus programas e vê-los sendo executados imediatamente, sem passar por fases intermediárias, como no caso das linguagens compiladas.

Infelizmente, o BASIC também apresenta algumas deficiências que se tornam progressivamente mais irritantes quanto mais complexas as aplicações que com ele queremos fazer. Talvez o ponto onde o BASIC se mostre mais inadequado seja no que diz respeito a sua estrutura interna. Um programa em BASIC é, necessariamente, um código linear, isto é, uma lista contínua de instruções onde a separação entre os diversos módulos funcionais é quase inexistente. Uma maneira de se verificar isso é observar que, para construir sub-rotinas ou coisa que o valha, só dispomos em BASIC da instrução GOSUB. Isto, convenhamos, não é suficiente.

PROGRAMAÇÃO ESTRUTURADA

Para que possamos compreender melhor esse argumento vale a pena fazer alguns comentários sobre o que chamamos hoje de **Programação Estruturada**. Este termo surgiu e foi adotado após o aparecimento de um documento de fundamental importância para o programador que leva seu trabalho a sério: o livro "Structured Programming", de autoria de Ole-Johan Dahl, Edsger W. Dijkstra e C. A. R. Hoare, considerados hoje em dia como verdadeiros "papas" no assunto. Este livro foi fruto de um debate muito intenso

O BASIC APRESENTA ALGUMAS DEFICIÊNCIAS EM APLICAÇÕES COMPLEXAS

ocorrido na Europa durante a década de 60. Nele encontramos o que poderíamos considerar como uma "disciplina" de programação. Trata-se, na verdade, de um apanhado de práticas, métodos e filosofias a serem seguidas ou evitadas durante a atividade de programação. Não pretendo, nem cabèria aqui, relacionar todos esses itens, no entanto, creio que um breve resumo será útil:

- Um programa deve deixar aparente a quem o lê todos os algoritmos e estruturas de dados utilizados durante sua confecção, isto é, **um bom programa é um programa claro.**
- Um programa deve, sempre que possível, refletir uma linha de raciocínio clara e direta, isto é, **um programa deve ter início, meio e fim.**
- Cada função específica a ser desempenhada pelo programa deve residir num módulo separado (**Filosofia modular**).
- O esforço de programação é reduzido quando cada pedaço (módulo) do programa é codificado em níveis de detalhamento progressivamente maiores, isto é, atacando primeiro os aspectos mais gerais do código e refinando-o seguidamente até obter o resultado final (Filosofia de refinamento por etapas ou, em inglês, **Stepwise Refinement**).

O PASCAL

Uma conseqüência prática do estabelecimento da filosofia de programação estruturada foi o aparecimento de linguagens de programação onde tais conceitos pudessem ser facilmente aplicados e que, por isso, foram chamadas de **linguagens estruturadas.** Dentre essas, podemos citar como exemplos típicos, o Algol-60, o Algol-68, o PL/I e o Pascal.

O Pascal, em particular, alcançou uma grande popularidade, principalmente entre os que lidam com pequenos computadores. A razão principal disso é simples: mais que qualquer uma das outras, o Pascal é uma linguagem **pequena**, o que é muito importante a fim de implementar a linguagem dispondo-se de uma quantidade limitada de memória, como no caso dos microcomputadores. Existem atualmente implementações dessa linguagem em virtualmente todos os tipos de computadores — até mesmo em máquinas grandes do tipo IBM, Burroughs, CDC etc.

A autoria da linguagem se deve ao Prof. Niklaus Wirth, da Escola Politécnica de Zurique, que a projetou no fim da década de 60. A primeira implementação — num computador CDC-6000 — ficou pronta em 1970, sendo que a descrição formal da linguagem foi editada em 1971.

O que o Prof. Wirth tinha em mente era uma linguagem apropriada para o ensino de programação que incorporasse os conceitos de Programação Estruturada. Esta meta, com efeito, foi alcançada e até mesmo ultrapassada, uma vez que hoje o Pascal é utilizado como instrumento não só de aprendizado como também de desenvolvimento de sistemas.

UM EXEMPLO

Para que o leitor possa avaliar melhor o que é **programar** em Pascal apresentamos a seguir um pequeno programa-exemplo. Para quem já programa em BASIC, colocamos também o mesmo programa escrito nessa linguagem.

O objetivo do programa é o seguinte: ele lê do teclado uma frase e, a seguir, conta quantas vezes cada letra do alfabeto apareceu. O usuário deverá digitar normalmente o texto a ser analisado e, em seguida, apertar a tecla "Return".

Programa em Pascal:

```
{                                    }
{  Programa para contar letras       }
{                                    }
PROGRAM contaletras (Input, Output);
    VAR
        ch : char;
        freq : ARRAY ['A' . . 'Z'] OF integer;
    BEGIN
        {Inicializo o ARRAY freq com zeros}
        FOR ch: = 'A' TO 'Z' DO freq [ch] : = 0;
        WRITELN ('Digite uma frase : ');
        {'Loop' para ler e contar as letras}
        REPEAT
            READ (ch);
        IF ch IN ['A' . . 'Z']
            THEN freq [ch] : = freq [ch] + 1
    UNTIL eoln;
    {Mostro os resultados}
        FOR ch : = 'A' TO 'Z' DO
            WRITELN ('Letra', ch, ' → ', freq [ch])
        END.{Fim do programa 'contaletras'}
```

Programa em BASIC:

```
10  REM
20  REM Programa para contar letras
30  REM
40  DIM FRQ (26)
45  REM Leio a Frase
50  PRINT "Digite uma frase:"
60  INPUT S$
65  REM 'Loop' para contar as letras
70  FOR I = 1 TO LEN (S$)
80  L$ = MID$ (S$, I, 1)
90  IF L$ < "A" OR L$ > "Z"
    THEN GOTO 120
```

```
100  J = ASC (L$) – ASC ("A") + 1
110  FRQ (J) = FRQ (J) + 1
120  NEXT I
125  REM Mostro o resultado
130  FOR I = 1 TO 26
140  PRINT "Letra"; CHR$ (I
     + ASC ("A") - 1; "→", FRQ (I)
150  NEXT I
155  REM Fim do programa
160  END
```

Resultado da execução: (o mesmo para as duas versões).

Digite uma frase:

THE QUICK BROWN FOX JUMPS
OVER THE LAZY DOGS

Letra A → 1	Letra N → 1
Letra B → 1	Letra O → 4
Letra C → 1	Letra P → 1
Letra D → 1	Letra Q → 1
Letra E → 3	Letra R → 2
Letra F → 1	Letra S → 1
Letra G → 1	Letra T → 2
Letra H → 2	Letra U → 2
Letra I → 1	Letra V → 1
Letra J → 1	Letra W → 1
Letra K → 1	Letra X → 1
Letra L → 1	Letra Y → 1
Letra M → 1	Letra Z → 1

ANALISANDO OS EXEMPLOS

A diferença mais chocante que podemos perceber quando comparamos os dois programas acima é no que se refere à disposição dos símbolos ao longo do texto. Enquanto que o programa em BASIC apresenta-se uniformemente ajustado à margem esquerda, as linhas do programa em Pascal mostram-se aparentemente "soltas".

Na verdade, quando vamos escrever programas em uma linguagem estruturada como o Pascal, a disposição das linhas ao longo do texto é de importância fundamental. O que fizemos no programa acima foi simplesmente aplicar uma regra que diz: "quando um comando, ou série de comandos, está subordinado a outro deve aparecer numa coluna mais à direita". Assim, por exemplo, os comandos "READ (ch)" e "IF ... THEN" apresentam-se em colunas à direita do comando "REPEAT"; do mesmo modo, a cláusula "THEN" do comando "IF" apresenta-se à direita deste, e assim por diante. Esta regra não faz parte da linguagem; na realidade, o Pascal não impõe quaisquer restrições quanto à disposição dos comandos nas linhas; trata-se apenas de uma regra de senso comum, universalmente adotada.

Um outro ponto de distinção que se pode perceber é no que diz respeito à maneira com que as duas linguagens encaram a utilização de variáveis. Em BASIC, basta citar o nome de uma variável para que a mesma seja criada, caso não exista anteriormente (apenas as variáveis dimensionadas precisam ser declaradas explicitamente, como a variável "FRQ" do programa acima). Por outro lado, o Pascal requer que **todas** as variáveis a serem utilizadas no decorrer do programa sejam previamente declaradas; isto é feito no nosso exemplo, nas linhas compreendidas entre a palavra "VAR" e "BEGIN". Essa estratégia é empregada por várias razões: ela permite que o programador tenha um ponto de referência, que o ajuda a manter-se a par das estruturas de dados empregadas no programa; ela força o programador a manter-se organizado, e, além do mais sem ela seria impossível definir novas estruturas de dados.

Um detalhe que talvez tenha passado despercebido é o fato de sermos obrigados em Pascal a **inicializar** todas as variáveis do programa antes de podermos fazer referência a elas. Isto pode ser notado no exemplo pela instrução

FOR ch : = 'A' TO 'Z' DO . . .

que aparece logo após a palavra "BEGIN". Através dela, todos os elementos do ARRAY "freq" são inicializados com o valor zero. Ora, isto é feito automaticamente em BASIC, por que não em Pascal? Esta, na verdade, é uma das omissões deliberadas da linguagem; ela força o programador a ter cuidado com os valores das variáveis — se tentássemos referenciar qualquer elemento daquele ARRAY sem termos atribuído previamente um valor qualquer a ele, isto acarretaria um **erro de execução.**

Existem ainda outros pontos que poderíamos abordar nesta discussão, mas o espaço não nos permite. Deixamos isto para o leitor (você notou, por exemplo, que os índices do ARRAY "freq" são **letras?**). Para ajudá-lo, coloquei abaixo uma breve descrição dos comandos Pascal utilizados no exemplo:

- Um texto compreendido entre os caracteres de abre e fecha-chaves é interpretado como um comentário em Pascal.
  O comando "FOR" é muito semelhante ao seu homônimo BASIC, isto é, consiste num comando de repetição onde uma variável de controle assume um valor que é incrementado a cada vez que o "loop" é repetido.

O PASCAL ALCANÇOU UMA GRANDE POPULARIDADE EM PEQUENOS COMPUTADORES

HOJE O PASCAL JÁ É USADO NO DESENVOLVIMENTO DE SISTEMAS

- O comando "WRITELN" é parecido com o "PRINT". Os valores entre parênteses são escritos numa unidade de saída.
- O comando "REPEAT" é, também, um comando de repetição. Ele se encarrega de repetir todos os comandos que figuram entre a palavra "REPEAT" e a palavra "UNTIL" até que a condição expressa após o "UNTIL" seja verdadeira.
- O comando "READ" é o equivalente Pascal do comando "INPUT" do BASIC, com a diferença que não é necessário apertar a tecla "Return" para efetivar a leitura. É possível detectar quando esta tecla foi apertada testando-se a condição "eoln" (End Of Line) que, no exemplo, aparece após a palavra "UNTIL".
- O comando "IF" também deve ser familiar ao programador "BASIC". Trata-se de um comando condicional onde se ordena que o comando expresso após a palavra "THEN" seja executado somente se a expressão condicional que figura entre o "IF" e o "THEN" for verdadeira (em Pascal, o comando "IF" pode ser escrito com uma segunda cláusula além do "THEN" — trata-se da cláusula "ELSE", que não foi usada no exemplo; algumas versões do BASIC também admitem esta extensão).

CONCLUSÃO

Mesmo depois de todo esse papo, ainda é difícil formar uma opinião. Antes disso seria preciso uma análise mais profunda das duas linguagens. Alerto os leitores para o seguinte fato: todas as vantagens do Pascal só se tornam evidentes quando precisamos fazer um programa realmente **grande**. Muitos aspectos importantes da linguagem foram desprezados nesta abordagem como, por exemplo, os mecanismos de subprogramação, as estruturas de dados, a característica recursiva da linguagem etc. Talvez o exemplo utilizado tenha sido inadequado, mas isso foi impossível evitar. Há também uma série de golpes baixos que se poderia fazer na versão em BASIC e que melhorariam a "performance" do exemplo. Nada disso, no entanto, é importante.

O que é fundamental é que o leitor encare seriamente o fato de que há alternativas para o BASIC. Não quero dizer com isso que o Pascal é a linguagem ideal para todos os casos; nenhuma linguagem o é. O Pascal é, apenas, mais uma possibilidade a considerar.

Para o leitor que me seguiu pacientemente até aqui, e que está interessado no assunto, anexei, além de uma bibliografia, uma lista de implementações da linguagem mais conhecida.

BIBLIOGRAFIA

Livros sobre linguagens em geral:

- *Programming* Language Concepts, de Carlo *Guezzi* e Mehdi Jazayeri — John Wiley & Sons.
- Programming Language Structures, de E.I. Organick, A. I. Forsythe e R. P. Plummer — Academic Press.

Livros sobre Pascal:

- Programming in Pascal, de Peter Grogono - Addison Wesley.
- A Primer on Pascal, de Richard Conway, David Gries e E. C. Zimmerman — Academic Press.
- Pascal User Manual and Report, de Kathleen Jensen e Niklaus Wirth — Springer Verlag.

Livros sobre programação estruturada:

- Programação Sistemática, de Niklaus Wirth.
- Algorithms + Data Structures = Programs, de Niklaus Wirth — Prentice-Hall.
- Structured Programming, de O. J. Dahl, E. W. Dijkstra e C. A. R. Hoare — Academic Press.

Livros sobre implementações da linguagem Pascal:

- The U. C. S. D. Pascal Handbook, de Randy Clark e Stephen Koehler-Prentice-Hall
- Pascal — The Language and Its Implementation, editada por D. W. Barron - John Wiley & Sons.
- The Pascal Handbook, de Jacques Thiberghien — Sybex.

IMPLEMENTAÇÕES MAIS CONHECIDAS DO PASCAL

U. C. S. D. Pascal — Implementação portável da linguagem feita pela Universidade da Califórnia, San Diego. Atualmente instalada em muitos minis e micros, inclusive os do tipo Apple, TRS-80 e IBM-PC. Informações: Softech MicroSystems - 9494 Black Mountain Road, Building 3, San Diego, California 92126, EUA.

Apple Pascal — Implementação da linguagem em computadores Apple. Informações: Apple Computer Inc., Advertising and Promotion Dept. — 20525 Mariani Avenue, Cupertino, CA 95014, EUA.

Pascal/Z — Implementação da linguagem em micros que utilizem o sistema operacional CP/M. Informações: Ithaca Intesystems, Inc — 1650 Hanshaw Road, P. O. Box 91 — Ithaca, New York 14850, EUA.

Pascal/MT+ — Pascal para CP/M. Informações: Lifeboat Associates — 1651 Third Avenue, New York, NY 10028, EUA.

Pascal/M — Pascal para CP/M; semelhante ao U. C. S. D. Pascal. Informações: Sorcim — 2273 Calle de Luna, Santa Clara, California 95035, EUA.

Pascal-S — Subconjunto da linguagem, definido por Niklaus Wirth. Uma listagem do compilador-interpretador que o processa pode ser obtida no livro "Pascal — The Language and Its Implementation", que figura na bibliografia. ■

consultoria

Trata-se de uma colúna mensal destinada a formar um elo entre o usuário e o fabricante de equipamentos abrindo-se ao leitor para recebimento de cartas que contenham perguntas sobre funcionamento, operação, preços, modificações e falhas em microcomputadores e periféricos de fabricação nacional ou importados.

Se você tem dúvidas sobre algum assunto ligado à informática, escreva para:

CONSULTORIA — REVISTA "INTERFACE" — ESTRADA DO TINDIBA, 2380 — JACAREPAGUÁ — RJ — CEP 22700.

P Sou usuário do microcomputador TK82-C. Como estou iniciando em programação BASIC, gostaria de obter algumas "dicas" para economizar memória nos meus programas.

R Apresentamos a seguir uma relação de "dicas" que poderão auxiliá-lo na redução de memória nos seus programas:

• Reduza o número de linhas do programa ao mínimo, pois cada linha no programa ocupa cinco bytes, assim distribuídos: dois bytes para o número da linha, dois para o comprimento e um para o término (new line).

• Retire as instruções REM e os prefácios explicativos.

• Elimine todos os resultados intermediários em cálculos. Por exemplo, A/(B*C) substitua por A/B/C.

• Reduza a uma mesma linha todas as operações aritméticas e lógicas possíveis.

• Só dimensione matrizes quando necessário; tente evitá-las.

• Utilize variável, ao invés de constantes numéricas, sempre que repetir um número mais que três vezes no programa, pois as variáveis ocupam cinco bytes cada acrescidas do número de caracteres do nome da variável, enquanto as constantes numéricas ocupam o número de dígitos da constante, incluindo o ponto decimal se houver, acrescido de seis bytes.

• Quando utilizar a expressão IF X = 0 THEN. . . substitua por IF NOT X THEN. . ., e economize uma constante numérica ou seis bytes.

• Substitua as constantes numéricas e/ou variáveis pelos valores das funções e códigos. Por exemplo, LET X = 0 ou LET X = 1 por LET X = PI-PI ou LET X = PI/PI.

• Quando houver expressões como:

```
10 LET X = 1
20 LET Y = 1
Substitua por:
10 LET X = 1
20 LET Y = X
```

E economize cinco bytes.

• Utilize PEEK e POKE para armazenar valores, sempre que possível.

• Substitua expressões como:

```
IF Y < 100 THEN LET Y = Y + 1
Por:
LET Y = Y + (Y < 100)
Ou ainda:
10 IF INKEY$ = "A" THEN LET Y = Y + 1
20 IF INKEY$ = "B" THEN LET Y = Y – 1
Por:
10 LET Y = Y + (1 AND INKEY$ = "A") –
   (1 AND INKEY$ = "B")
Ou:
10 LET Y = Y + (INKEY$ = "A") –
   (INKEY$ = "B")
```

P Possuo um microcomputador DGT-100 e gostaria de saber como ler fitas gravadas em 500 bps do CP 500.

R O microcomputador DGT-100 possui um circuito interno, fornecido de fábrica, que lhe permite ler fitas gravadas em 500 bps.

Se o programa for em BASIC, faça o seguinte:

1 — Conecte o MIC do gravador ao MIC2 do computador.

2 — Conecte o MONITOR do gravador ao REM2 do computador. Como o segundo gravador nunca é acessado para leitura de programas, em 500 bps usamos o REM2 como entrada do circuito para fitas gravadas em 500 bps.

3 — Escreva CLOAD# $, e aperte RETURN. Note que existe uma vírgula após o $. Proceda a partir deste ponto, como se fosse um programa em 2000 bps.

Se o programa for em linguagem de máquina, faça as mesmas conexões descritas acima e:

1 — Escreva SYSTEM e aperte RETURN.

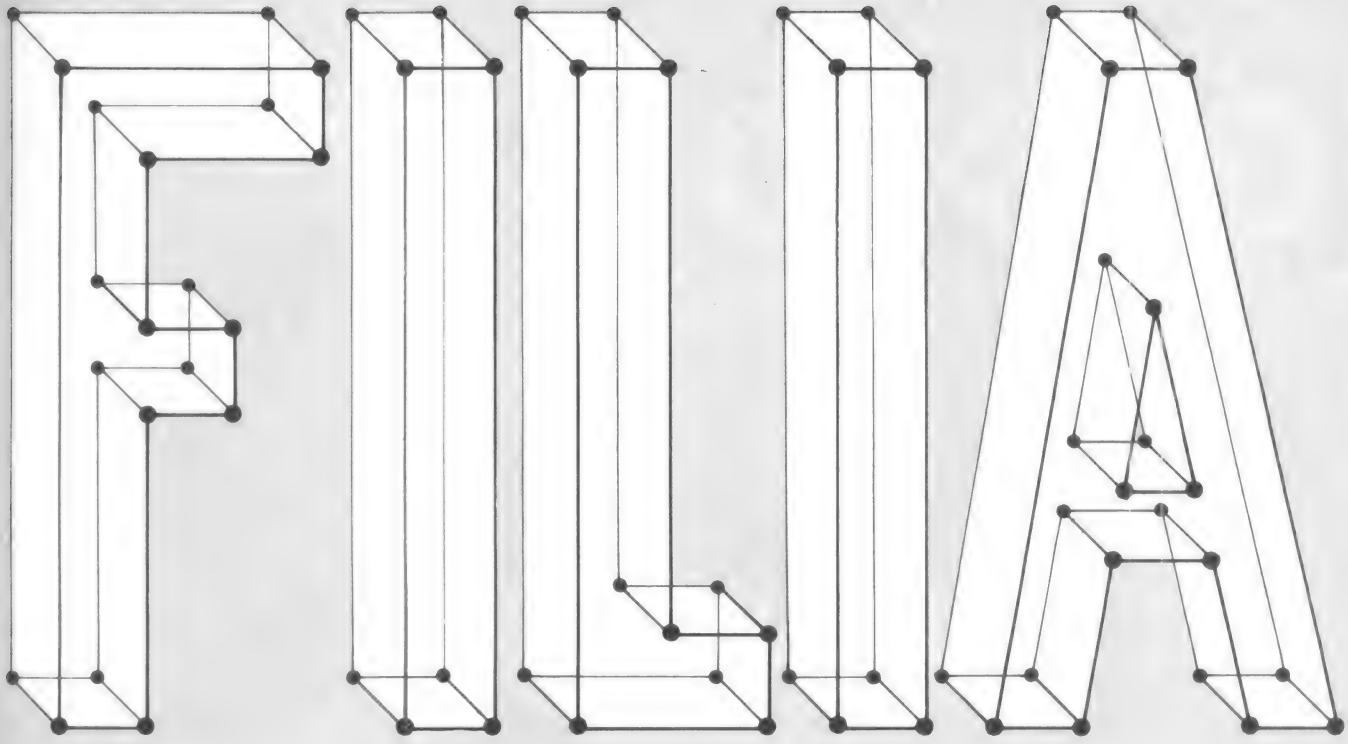
2 — Um * ? aparecerá no vídeo. Coloque então o nome do programa precedido de # $,. Note a vírgula depois do $. Aperte RETURN.

Por exemplo:

$, TESTE (programa chamado TESTE será lido em 500 bps)

3 — Proceda a partir deste momento como se fosse um programa em 200 bps.

Se você for ler dados gravados, diretamente, em 500 bps, faça as mesmas conexões descritas acima e use o comando: INPUT # $,. Note a vírgula depois do $.



UM SISTEMA DE PROGRAMAÇÃO PARA MICROCOMPUTADORES

3.ª Parte

Por Rildo Pragana

Filia possui dois interpretadores distintos: o interpretador externo, responsável pela interação com o programador, que aceita comandos como "apague a última tecla introduzida" ou "execute esta linha de comandos", e o interpretador interno que implementa o **código alinhavado**. Nessa parte do artigo discutiremos principalmente o interpretador interno e sua implementação para o 8080 ou Z80.

Para entendermos melhor como funciona o código alinhavado, imagine que você desenvolve seus programas como uma série de sub-rotinas padronizadas e a certo momento escreve um programa de "alto-nível", consistindo apenas em "CALL's" para as sub-rotinas já desenvolvidas. O programa escrito assim consistiria num aglomerado de chamadas de sub-rotinas que resolvem o problema, sem que seja necessário repetir as funções já implementadas por aquelas sub-rotinas, resultando numa forma primitiva de código alinhavado.

Evidentemente, notamos a presença de muitos códigos de "CALL" que poderiam ser eliminados pois somente os endereços das sub-rotinas trazem informação relevante. Retirando os "call's" e escrevendo um interpretador para chamar cada sub-rotina, dados os seus endereços (ponteiros), teríamos o **código alinhavado direto**. A modalidade de nossa escolha recai no código alinhavado indireto devido ao tratamento homogêno das rotinas escritas em "alto-nível" e das escritas em linguagem máquina. No **código alinhavado indireto** os programas consistem em ponteiros indiretos (isto é, que indicam outros ponteiros) para o código a ser executado. Cada ponteiro indireto ou **descritor** aponta para um segundo ponteiro que pode ser: 1) o ponteiro onde começa a rotina caso ela seja escrita em linguagem máquina do processador; 2) o endereço de uma rotina de sistema que define a classe ou família de funções à qual o átomo em questão pertence.

Assim, todos os programas começados por dois pontos ":" possuem um ponteiro para a rotina que implementa um "call" em alto-nível. Todas as constantes têm um ponteiro para a parte executável do compilador CONST, as variáveis o mesmo em relação ao compilador VAR, e assim por diante.

A tarefa do interpretador interno é executar o código apontado indiretamente pelo **descritor** e manter um **contador de programa** semelhante ao que existe no processador (às pessoas não familiarizadas com programação em assembly recomenda-se a leitura de um dos muitos manuais existentes para esse fim, se desejarem entender completamente a operação do interpretador).

Na nossa implementação o pseudo-PC, isto é, o contador de programas do interpretador interno, é o par de registradores BC do 8080/Z80. Os programas em linguagem máquina deverão ser escritos de modo a retornar o valor antes existente nesse par de registradores para que Filia possa operar satisfatoriamente. Outra curiosidade é o uso do SP (ponteiro da pilha) como ponteiro para a pilha de usuário e não de sub-rotinas, como é usado normalmente. O retorno ao Filia é conseguido pelo salto ao interpretador interno e não com o RET familiar como o é para a maioria de vocês. As chamadas de sub-rotinas em Filia empilham o endereço de retorno numa segunda pilha cujo ponteiro é uma variável de nome SS (dois bytes na memória), com acesso limitado ao usuário.

O INTERPRETADOR INTERNO

Mostraremos agora como opera o interpretador interno, chamado PROX porque ele é usado para interpretar o próximo átomo em um programa.

A listagem dada é em linguagem assembly da Intel mas serve também para o processador Z-80 da Zilog. Essa forma foi escolhida porque o assembler residente do Filia será dado com mnemônicos da Intel, originariamente usados com o processador 8080. Com referência, leia-se o manual "8080 Assembly Language programming" editado pela Intel.

```
PROX:   LDAX   B        ;  leia o descritor apontado pelo pseudo-PC
        MOV    L, A     ;  deixando-o em HL. Primeiro
                        ;  o byte menos significativo
        INX    B        ;  avance o pseudo-PC
        LDAX   B        ;  agora leia o byte mais significativo
        MOV    H, A     ;  descritor agora em HL
        INX    B        ;  pseudo-PC aponta para a próxima "instrução"
        MOV    E, M     ;  leia agora o endereço de execução
                        ;  começando pelo byte menos significativo
        INX    H        ;  agora o byte mais significativo
        MOV    D, M     ;  em DE o endereço de execução
        XCHG            ;  passe esse endereço para HL
        PCHL            ;  Execute!
```

Podemos observar que o pseudo-PC (par BC) retorna do interpretador interno apontando para a próxima "instrução" de alto-nível a ser interpretada.

O par DE também contém informações relevantes sobre o ponteiro descritor que foi procurado, fato que será posteriormente utilizado. Como A, D, E, H e L foram destruídos, não é possível passar parâmetros para sub-rotinas usando esses registradores. Na verdade, em Filia quase sempre passamos parâmetros na pilha, mesmo para as rotinas em linguagem assembly.

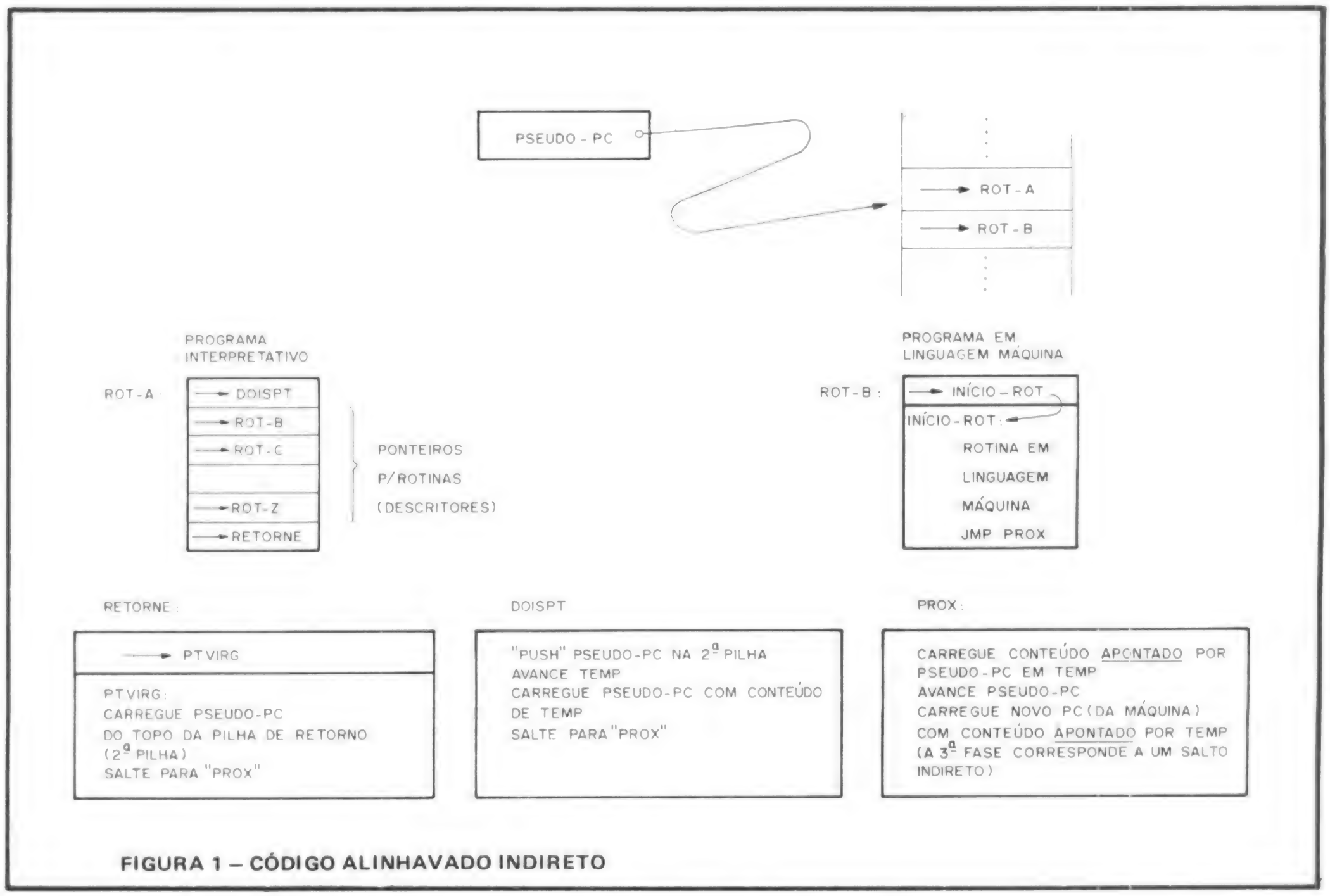


FIGURA 1 – CÓDIGO ALINHAVADO INDIRETO

O MICRO QUE JÁ NASCEU GRANDE

CP-300

CP-500 · CP-300 · CP-200

CARACTERISTICAS TÉCNICAS

- CPU com microprocessador Z80 de 2MHz memória principal de 48KB.
- Teclado alfanumérico com 54 teclas e função auto repeat-reset com proteção contra acionamento involuntário.
- Conexões — para TV ou monitor de vídeo — 16 linhas com 64 col. — 16 linhas com 32 col. - modo gráfico com 48 x 128 pontos.
- Interface para gravador com velocidade de 500 e 1500 Bauds.
- Expansões para: controlador de drives, porta paralela (impressora) e porta serial RS 232 (Modem, etc.).
- Dimensões: altura: 7,5 cm — compr.: 33,0 cm — larg.: 22,0 cm — peso: 1,5 kg.

Compatível em todos os aspectos com o CP-500, inclusive no software, porém é composto de módulos que podem ser adquiridos de acordo com suas necessidades.

SOFTWARE

JOGOS:
BATALHA AÉREA
- CP-200 (F) . 3.850,00
- CP-500 (F/D) 4.400,00*

BATALHA NAVAL
- CP-200 (F) . 5.830,00

FORCA
- CP-200 (F) . 4.400,00
- CP-500 (F/D) 4.620,00*

TIRO AO ALVO
- CP-200 (F) . 3.850,00
- CP-500 (F/D) 4.400,00*

BIORRITMO
- CP-200 (F) . 3.850,00

LOTO
- CP-200 (F) . 4.180,00

TABUADA
- CP-200 (F) . 3.850,00
- CP-500 (F/D) 3.080,00*

SIMULADOR DE VÔO
- CP-200 (F) . 6.600,00

COMANDO UFO
- CP-200 (F) . 3.850,00
- CP-500 (F/D) 4.950,00*

OESTE SELVAGEM
- CP-200 (F) . 4.620,00

SENHA
- CP-200 (F) . 4.950,00

INVASÃO CÓSMICA
- CP-200 (F) . 6.050,00

SÓMENTE P/CP-500

PATRULHA (F/D)
6.160,00*

INVASORES (F/D)
5.610,00*

PADDLE PINBALL (F)
Simula jogo de fliperama
9.600,00

DISCOS VOADORES (F)
Controle o canhão para abatê-los 8.000,00

DANCING DEMOND (F/D) - Incrível demoninho dançarino . . 8.000,00*

XADREZ (F/D)
6 níveis que vão desafiá-lo 10.000,00*

CUBO (F/D)
Você nunca resolveu o cubo? A solução é cubo 6.400,00*

JOGOS EM BASIC (F/D) 6.400,00*

JOGOS (F/D) COSMIC, SCARFMAN, LUNAR, BARRICADA, GALAXI, METEOR, PENETREITOR - cada . 9.600,00*

PROMOÇÕES
PACOTE ECONÔMICO (F)
Para CP-200 NE-Z8000, com música, damas, gamão, conta corrente e controle de estoque 4.000,00

10 JOGOS EM DISCO
Para CP-500, variedade: boa, sky, pouso lunar, jornada, teaser, cupim, hopper, cram, fireman, spacefire . . . 24.000,00

APLICATIVOS
SOMENTE P/CP-200 (F)

CONTAS A PAGAR
Controla o acumulado do mês, ano e operações e poupança . . 13.200,00

AGENDA
Trabalha com dois tipos de informação atividade e telefone 8.800,00

CADASTRO DE CLIENTE
Cadastra p/cliente a razão social, rua, bairro, cid., est., CEP., tel., produto adquirido, data da aquisição 13.750,00

VÍDEO-TÍTULO
Cria títulos para vídeo-tapes, inclusive com movimentos 16.500,00

VU-CALC
Destina-se à execução de cálculos sobre uma planilha 10.000,00

SOMENTE P/CP-500

CADASTRO DE CLIENTES (D) . . . 20 ORTN

MALA-DIRETA (D/I)
Sistema de mala direta permitindo a impressão de etiquetas utilizando ou não chave de acesso 40.000,00

FINANÇAS (D)
Engloba funções de juros compostos, análise de preço de venda e rendimentos, taxas de retorno, tabela de amortização, saldo hipotecário, pagamentos a prazo . 40.000,00

PROCALC (D)
Destina-se a execução de cálculos sobre uma planilha 120.000,00

VÍDEO (F/D)
Editor gráfico da tela. Desenhe com facilidade no CP-500. Armazena as telas em fita ou diskette 10.000,00*

BANNER (F/D/I)
Imprime mensagens em letras garrafais (80 col.) 6.400,00*

SCRIPY (I)
Completo e versátil processador de texto . . . . . em fita . . . . 32.000,00 em disco . . . 40.000,00

CONTROLE DE AÇÕES (D/F)
Para quem gosta da Bolsa. Mostra os resultados (Lucro/Perda) 6.400,00*

DIRETÓRIO (D)
Organiza e cadastra todos os seus programas e arquivos em diskette automaticamente. Pesquisa por disco ou programa e pode imprimir ordenadamente 16.000,00

BANCO DE DADOS (D/I)
Sistema de fichário eletrônico. Você mesmo cria as fichas e pode pesquisar de vários modos, além de permitir impressão. Similar ao Profile 40.000,00

CARTA ASTRAL (F/I)
Este programa faz todos os cálculos necessários ao desenho de uma carta astral e ainda imprime a mesma em questão de minutos. Só não interpreta 15.000,00

UTILITÁRIOS
(SOMENTE P/CP-500)

CONVERT (F/D)
Converte números decimais e hexa. Pode ser chamado a qualquer instante e não atrapalha o BASIC 6.400,00*

LISTA (D/I)
Imprime as listagens de programas em BASIC de forma limpa e organizada. Não se perca com LLIST 16.000,00

SUPERTECLA (F)
Cada tecla de seu CP-500 representa duas palavras do BASIC além de seu valor normal. Reduz drasticamente o tempo gasto na teclagem de programas 8.000,00

ODONTO (F/D)
Engloba setor financeiro, agenda, cadastro, mala direta, lay-out dentário . . . . . 50 ORTN

EDITOR (D)
Poderoso editor assembler para os que programam em linguagem de máquina. Manual c/instruções 32.000,00

SOUND (F/D)
Pequena mas útil subrotina que cria uma nova palavra em BASIC — "SOUND". Instruções detalhadas de como colocá-la em seus programas 6.400,00*

F- Para CP-500/CP-300 ou CP-200 em fita. I - Requer impressora.
D- Para CP-500/CP-300 em disco. * - Acrescentar Cr$ 6.000,00 p/versão em disco

filcres

Filcres Importação e Representações Ltda.
Rua Aurora, 165 — CEP 01209 — São Paulo — SP
Telex 1131298 FILG BR — PBX 223-7388 — Ramais 2, 4, 12, 18, 19 — Diretos: 223-1446, 222-3458, 220-5794 e 220-9113 - Reembolso - R17 Direto: 222-0016, 220-7718

NOME . . . . . TEL. INT8
END. . . . . CEP . . . . .
CID. . . . . EST. . . . . EQUIP. . . . .
ENVIE AS INFORMAÇÕES ACIMA PARA FILCRES IMP. E REPR. LTDA.

Chamadas de sub-rotina em alto-nível necessitam que o pseudo-PC seja salvo na segunda pilha e então carregado com seu novo valor. Como no 8080 só existe uma pilha implementada por hardware, esta segunda pilha tem como ponteiro uma localidade na memória SS, e os "push" e "pop" devem ser realizados por seqüências que simulem a operação de uma pilha. A rotina responsável pelo "call" é chamada de **doispt**:

```
DOISPT:  LHLD   SS      ;  ponteiro da 2ª pilha em HL
         DCX    H       ;  prepare para um "push" no
                           pseudo-PC
         MOV    M, B    ;  mova o mais significativo
                           byte
         DCX    H       ;  atualize o ponteiro
         MOV    M, C    ;  e mova o outro byte
         SHLD   SS      ;  guarde de volta o ponteiro
                           da 2ª pilha
         INX    D       ;  corrija endereço do próximo
                           item a interpretar, na sub-ro-
                           tina
         MOV    C, E    ;  mova esse endereço para o
                           pseudo-PC
         MOV    B, D
         JMP    PROX    ;  e interprete
```

Em síntese, o DOISPT empilha na 2ª pilha o conteúdo do pseudo-PC e "salta" para a instrução seguinte da sub-rotina. Observe que a chamada de sub-rotina é, de certo modo, contida no início da própria sub-rotina. Para entender o que estava no par DE e que foi passado para BC na rotina DOISPT, revise novamente o código de PROX. Veja que em DE ficou o descritor da sub-rotina incrementado de 1 byte. Eis porque, incrementando-o mais um byte teremos este apontando para a primeira instrução em alto-nível da sub-rotina que iniciamos na execução.

O retorno de uma procedure ou sub-rotina é mais simples: basta retirarmos o novo valor do pseudo-PC do topo da segunda pilha. Essa rotina é chamada de PTVIRG (abreviatura de ponto-e-vírgula).

```
PTVIRG:  . DW    $ + 2   ;  apontador para o início da
                            rotina, pois esta é interpreta-
                            da por PROX
         LHLD    SS      ;  faremos um "pop" do PC na
                            segunda pilha
         MOV     C, M    ;  na ordem inversa de entrada
         INX     H       ;  avance o ponteiro
         MOV     B, M    ;  agora o mais significativo
         INX     H       ;  deixe o ponteiro atualizado
         SHLD    SS      ;  . . . guardando-o de volta
         JMP     PROX    ;  interprete com o novo valor
                            do pseudo-PC
```

A maior vantagem de se usar o código alinhavado indireto nessa implementação é a simplicidade com que os compiladores podem ser escritos. Um compilador (programa que traduz texto em códigos) para código alinhavado indireto só precisa produzir ponteiros, sendo portanto independente do processador usado (para ser dependente ele teria que produzir código-máquina do processador específico). Isso acontece também com Pascal, que usa normalmente o código-p como máquina virtual intermediária. A maior desvantagem do código alinhavado indireto sobre os demais é o tempo de processador consumido para interpretar cada código (ponteiro), mas isso é suplantado pela flexibilidade na compilação.

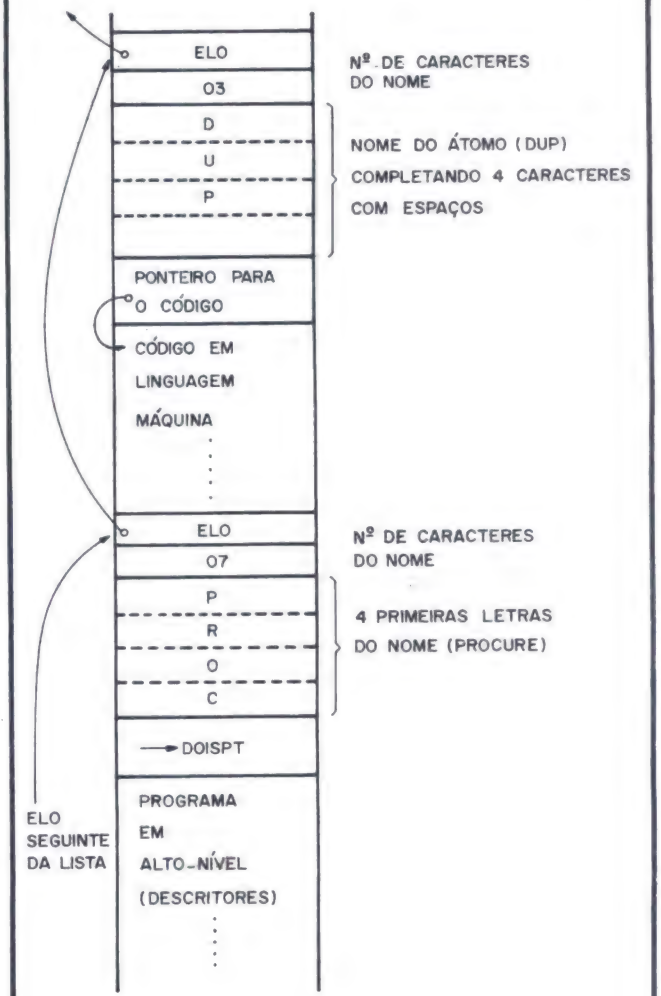


FIGURA 2 – ORGANIZAÇÃO DO DICIONÁRIO

ORGANIZAÇÃO DO DICIONÁRIO EM FILIA

O dicionário em Filia (assim como em Forth) é uma estrutura linear onde cada átomo representa uma entrada com ligação (link) numa lista de átomos existentes, os quatro primeiros caracteres do seu nome e o número total de caracteres contidos no nome, além do ponteiro para a rotina que deverá ser executada na sua evocação. Somando os bytes gastos com cada item desse cabeçalho, teremos:

ligação ou elo (link)	2 bytes
nome	4 bytes
tamanho do nome	1 byte
ponteiro para código (descritor)	2 bytes
Total	9 bytes

Esse é o "overhead" de memória gasto na definição de cada átomo de Filia. Uma variável, por exemplo, gastaria 9 + 2 bytes, os últimos dois bytes para armazenar o conteúdo da variável. Um programa em Filia tem tamanho variável,

mas em média são gastos 2 bytes por cada outro átomo evocado na sua definição, o que pode resultar em programas muito mais compactos que os escritos até mesmo em linguagem assembly! A propósito, podemos também usar o assembler residente no sistema para codificar programas que serão traduzidos para linguagem máquina, e também nestes serão incluídos os 9 bytes extras de cabeçalho.

Uma pergunta resta no ar: como o interpretador (externo, no caso) encontra uma função (átomo) pelo nome? A resposta é simples. Os dois bytes de ligação da última definição no dicionário estão contidos (apontados) pela variável de sistema HEAD (como no Forth). Estes, por sua vez, apontam para a ligação do penúltimo átomo definido, que aponta para o antepenúltimo . . .

O final (isto é, o primeiro átomo definido quando o dicionário foi criado) contém 0 (zero), que indica o fim da lista.

O uso de uma lista ligada é necessária por causa do tamanho variável ocupado por cada átomo. A rotina PROCURE é responsável por varrer a lista à procura de um nome que coincida com o nome do comando que desejamos executar. Uma observação muito importante deve ser feita. Como são armazenados apenas os quatro primeiros caracteres do nome e o número de caracteres do mesmo, dois nomes diferentes para serem identificados como tal deverão ter algumas diferenças nas quatro primeiras letras, ou possuir tamanhos diferentes. De outra forma, PROCURE os identificaria como o mesmo átomo. Quando redefinimos uma função (com o mesmo nome, fazendo tarefas diferentes) somente a última definição é considerada pelo sistema. Isso reflete a ordem com que PROCURE faz a pesquisa dos nomes: do último no dicionário para trás.

A rotina PROCURE devolve no topo da pilha um flag indicando se encontrou ou não o átomo pesquisado e, caso afirmativo, também o endereço de execução (isto é, o ponteiro descritor) da rotina. Assim, já podemos dizer como a função (apóstrofo) funciona:

```
IMED  ÁTOMO  PROCURE  APAG  2  +  ;
```

Algumas funções são novas para nós, mas as explicaremos rapidamente. IMED tem a finalidade de transformar a função definida (no momento ') como "imediata", isto é, que executa mesmo durante uma compilação. A função ÁTOMO obtém do buffer de recepção (onde estão os caracteres introduzidos via teclado) um "token" ou "átomo", separados pelo delimitador usual, que é o espaço.

A função definida como a de cima tem uma deficiência. Não indica quando a pesquisa não foi bem sucedida. Para corrigir este detalhe, devemos introduzir um condicional que verifique o sucesso da busca de PROCURE e aborte a execução quando o átomo pesquisado não existir. A nova solução será:

```
IMED  ÁTOMO  PROCURE  SE  (se encontrou . . .)
   2 +                        (some 2 ao endereço do descritor)
      SENÃO (não encontrou, aborte!)
   ABORTE
            ENTÃO;
```

A função PROCURE pode ser usada também quando da criação de uma nova função para verificar se não existe outra com o mesmo nome, fornecendo ao operador uma mensagem de advertência. Isso porque, quando redefinimos uma função perdemos acesso à antiga (de mesmo nome, evidentemente).

Outra função de muita utilidade é o EXEC (abreviatura de execute) que permite executarmos uma função dado o seu descritor na pilha. Observe que com 'teremos 2 bytes de diferença (a mais) em relação ao descritor. Por exemplo, fazendo' APAG 2 – EXEC é equivalente a fazer simplesmente APAG.

Os mais curiosos devem estar procurando entender "prá que serve isso".

A potência desta operação reside em passarmos parâmetros para sub-rotinas de uma maneira totalmente diferente. Em geral temos sub-rotinas que esperam os **valores**, como + por exemplo. Outras esperam como parâmetro o **endereço** de um operando, por exemplo, a função = . Agora podemos passar como parâmetro também um **descritor** de outra função. Uma utilidade a título de demonstração é no redirecionamento de entrada e saída. Considere que 10C seja uma rotina idêntica a 0C, mas que envie caracteres para outro dispositivo que não seja o terminal de vídeo (ou monitor, dependendo do sistema).

A seguinte rotina pode ser usada para imprimir os caracteres A a Z em qualquer saída desejada.

```
: TESTE-TERMINAL 5A 41 CONTE I SEG EXEC OUTRO APAG;
```

Para sua utilização, devemos primeiro fornecer a rotina a ser executada no seu interior para enviar um caractere. Fazendo ' 0C 2– TESTE-TERMINAL teremos a saída no nosso monitor de vídeo habitual. Se, por outro lado, fizéssemos ' 10C 2– TESTE-TERMINAL teríamos os caracteres impressos no local que 10C imprime (envia) um caractere. Em outras palavras, a função TESTE-TERMINAL precisa, como parâmetro, de uma **rotina** que deverá ser executada. Para os conhecedores de Algol é como uma passagem de parâmetros pelo nome, que em geral é causa de tanta confusão. O 2– que aparece nas linhas acima é para compensar o endereço (dois bytes a mais) que o ' devolve. O ' foi escrito para devolver o início dos parâmetros de uma estrutura de dados (variável, constante, array etc.), que vem logo após o descritor. Nos casos acima queremos o próprio descritor, daí subtrairmos sempre dois bytes.

Como o EXEC tem uma relação muito íntima com o interpretador interno, daremos também seu código em assembly agora:

```
EXEC:   .DW     $+ 2     ;  ponteiro para o início da rotina, abaixo
        POP     H        ;  obtenha o descritor da pilha
        MOV     E, M     ;  leia endereço de execução (como em PROX)
        INX     H
        MOV     D, M
        XCHG
        PCHL
```

Compare seu código com o de PROX para verificar a semelhança existente e depois veja se entendeu melhor como PROX e EXEC funcionam.

Veremos agora como uma constante numérica introduzida no interior de um programa é executada, ou seja, tem seu valor empilhado durante a execução do programa já compilado. Existe uma rotina chamada NÚMERO_COMPILADO que tem estreita relação com o interpretador interno e cuja função é processar o número que vem em seguida, no dicionário, introduzindo-o na pilha.

O acesso ao número é imediato pois ele está numa localidade apontada diretamente pelo pseudo-PC quando NÚMERO_COMPILADO executa. É necessário, porém, que o pseudo-PC seja avançado para que o interpretador interno não tente executar a constante numérica como sendo o descritor de uma rotina.

Essa mesma técnica é usada quando temos constantes "string" que devemos imprimir na execução de um programa. Estas são compiladas pelo átomo ." que imerje (compila) no dicionário a função equivalente a NÚMERO_COMPILADO em relação a strings: (. ") (leia-se ponto _ aspas entre parênteses).

A implementação de NÚMERO_COMPILADO é dada em seguida:

```
NÚMERO_COMPILADO: . DW    $ + 2  ;  ponteiro para o início da rotina
                  LDAX    B      ;  leia o primeiro byte (LSB) do número
                  MOV     L, A   ;  deixando em L
                  INX     B      ;  avance o pseudo-PC
                  LDAX    B      ;  agora o MSByte
                  MOV     H, A   ;  número (16 bits) em HL
                  INX     B      ;  deixe o pseudo-PC
                                 ;  apontando para a
                                 ;  próxima instrução
                                 ;  a interpretar
                  PUSH    H      ;  empilhe o número lido
                  JMP     PROX   ;  volte ao interpretador
```

Apesar do número, isto é, a constante numérica, ficar colocado junto ao código alinhavado, o pseudo-PC é avançado sobre o mesmo durante a execução para que o interpretador não tente decodificar esse número como uma instrução de Filia. Isso é mais ou menos o que acontece com o microprocessador em instruções (em máquina) de múltiplos bytes, quando o PC da máquina é avançado para ler os bytes de endereço ou dados, sem que o decodificador de instruções do processador tente executar esses dados como código de instrução.

Apesar de Filia ser estruturada em alto-nível, devemos ter mecanismos de saltos condicionais e incondicionais que deverão ser executados na interpretação do código alinhavado, para produzir as seqüências de fluxo de programa convenientes. Durante a compilação, as rotinas SE. SENÃO, ENTÃO, INÍCIO, ATÉ, REPITA etc produzem saltos dessa natureza que pré-formam a operação final esperada. As duas rotinas mais comuns para suprirem essa necessidade são XSE e XREP. A primeira faz um salto condicional para o endereço dado em seguida, quando o topo da pilha é um flag de valor zero. A segunda efetua um salto incondicional para o endereço dado em seguida.

```
XREP:    . DW    $ + 2
XREP1:   LDAX    B      ;  leia o endereço de "salto" dado no dicionário
         MOV     E, A   ;  . . . em DE
         INX     B
         LDAX    B
         MOV     B, A   ;  será o novo conteúdo do pseudo-PC
         MOV     C, E
         JMP     PROX   ;  volte ao interpretador

XSE:     . DW    $ + 2
         POP     H      ;  obtenha em HL o flag
         MOV     A, L   ;  somente L interessa
         ANA     A      ;  "and" para ligar flags (teste)
         JZ      XREP1  ;  flag é zero? se for dê o salto
         INX     B      ;  avance o pseudo-PC para não interpretar
         INX     B      ;  . . . o endereço de salto como instrução
         JMP     PROX
```

A rotina XSE dá um salto condicional, dependente do valor do flag contido no topo da pilha. Como esse flag assume valores um ou zero, apesar de ocupar 16 bits ou 2 bytes, somente a parte menos significativa do número precisa ser testada. Quando o flag for zero o salto deverá ser efetuado, daí a execução de XREP1. Quando o salto não for dado, mesmo assim devemos avançar o pseudo-PC para evitar que o interpretador interno (PROX) tente executar a "rotina" que, no caso, é o endereço de salto e que deve ser entendido como fazendo parte da instrução de salto XSE.

Evidentemente que esse interpretador de Filia ainda não está completo, mas a sua codificação pode ser iniciada agora. Aconselhamos aos nossos leitores um exame detalhado do seu funcionamento e não uma tradução cega dos mnemônicos em linguagem máquina, para ser possível uma adaptação melhor ao tipo de microcomputador disponível. Além disso, muitas idéias de implementação que serão úteis na criação de novas funções podem ser encontradas nos mecanismos dados agora. O próximo mês nos trará a implementação das estruturas de controle já vistas, e algumas considerações sobre o interpretador externo. ■

FORMAMOS UM GRUPO DE EMPRESAS PARA ACOMPANHAR A
RÁPIDA EVOLUÇÃO DA INFORMÁTICA.
CONTE CONOSCO E SAIBA POR QUE SOMOS "A SOLUÇÃO TOTAL EM INFORMÁTICA".

ADVANCING COMPUTER SHOP – ADVANCING SOFTWARE HOUSE Rua Sarmento Leite, 248
Porto Alegre – RS ☎26-0194 e 26-1194 ADVANCING CONSULTORIA DE PESSOAL E TREINAMENTO
EM INFORMÁTICA LTDA Rua dos Andradas, 1560 – Gal. Malcon – 5°and. – Cj 518 – P. Alegre – RS ☎ 26-8246

POLY 201 DATA ENTRY Segurança absoluta na entrada de dados.

Fala com a linguagem Cobol, fácil de operar como um telefone.

O POLY 201 DE foi concebido para você resolver com grande facilidade e agilidade a coleta, entrada e verificação de dados. Com ele você desafoga o computador central, liberando-o da tarefa de entrada de dados, disciplinando e otimizando ainda mais sua utilização.

Características básicas do POLY 201 DE

O POLY 201 DE dispõe de linguagem POLY DEL, linguagem de alto nível que fornece ao usuário enormes facilidades de programação, além de poder-se linkeditar com programas COBOL.

A linguagem POLY DEL é composta de seis módulos básicos:

- Módulo selecionador de estado
- Módulo de formatação de telas (programação)
- Módulos de tratamento de arquivo de dados sem controle de formatos
- Módulo de execução de pesquisa, mediante chave
- Módulo de entrada de dados onde é permitida a digitação, alteração, inserção e deleção de caracteres, avançar e recuar campos, gravar, atualizar, inserir e deletar registros
- Módulo de verificação de dados.

Possui um controle total de produtividade do operador.

Com 64 kb de RAM,utiliza-se de discos flexíveis como suporte magnético, podendo-se conectar unidades de fita magnética e impressoras de várias velocidades.

Dispõe de recursos de comunicação de dados através de ligações síncronas ou assíncronas, conectando-se a diversos computadores.

Possui teclado especialmente projetado para a entrada de dados.

POLY 201 DE, uma grande conquista na entrada e processamento de dados.

APU
ASSISTÊNCIA PRÉ-PÓS VENDA
SISTEMAS E SERVIÇOS POLYMAX

A Polymax presta a sua homenagem a todos os homens que fizeram conquistas geniais, melhorando a qualidade de vida da humanidade. Como Graham Bell, o pai do telefone.

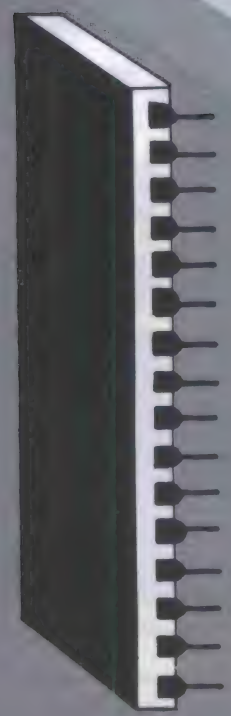
Polymax
SISTEMAS E PERIFÉRICOS S.A.
Uma grande conquista

MATRIZ: PORTO ALEGRE (RS) - FONE: 42-7833
FILIAIS: ABC (SP) - FONE: 454-4922 - **BELO HORIZONTE** (MG) - **BRASILIA** (DF) - FONE: 225-1456 - **CURITIBA** (PR) - FONE: 33-6525 - **PORTO ALEGRE** (RS) - FONE: 42-3311 - **RIO DE JANEIRO** (RJ) - FONE: 252-8274 - **SÃO PAULO** (SP) - FONE: 283-3722

REDE DE REVENDEDORES: BAURU (SP) - SOMA - 24-2558 - **BELÉM** (PA) - PAM - 222-9772 - **BELO HORIZONTE** (MG) - PROLOG - 337-5766 - **CAMPINAS** (SP) - MICROSYSTEMS - 51-3542 - **CAMPO GRANDE** (MS) - NEW LINE - 624-5349 - **CURITIBA** (PR) - DREHER - 224-0743 - **DUQUE DE CAXIAS** (RJ) - CPM - 771-0312 - **FORTALEZA** (CE) - GENERAL DATA - 226-2610 - **GOIÂNIA** (GO) - GENDADOS - 224-5486 / 224-5487 - **ITAJAÍ** (SC) - ENTEC - 44-0244 - **JUIZ DE FORA** (MG) - VERMAQ - 212-3809 / 213-5145 - **MANAUS** (AM) - CPD - 237-1033 / 237-1793 - **PIRACICABA** (SP) - SOGEMEC - 34-2100 - **RECIFE** (PE) - GENERAL DATA - 222-0357 / 222-2630 - **RESENDE** (RJ) - MICRO MAXI - 54-1644 / 54-2031 - **RIBEIRÃO PRETO** (SP) - KEHDI - 634-4715 - **RIO DE JANEIRO** (RJ) - EMBRASE - 222-4890 / 224-4379 - **SALVADOR** (BA) - SISMAQ - 243-0991 / 241-2894 - **SANTOS** (SP) - SISPER - 37-4705 - **SÃO JOSÉ DOS CAMPOS** (SP) - SISCON - 23-3752 - **SÃO LUIS** (MA) - M.P.A. - 221-1755 - **SÃO PAULO** (SP) - DATALOG - 211-9202 - **UBERLÂNDIA** (MG) - SILOGICA - 234-4191 - **VOLTA REDONDA** (RJ) - BEVORELLI - 42-2175 - SISCON - 42-1412.

Associado à ABICOMP

CURSO Z-80



9ª Lição

Por André Gil Rubens e Ney Acyr de Oliveira
(A & N – Consultoria em Hardware e Software de Microcomputadores)

Estudaremos nesta lição as vantagens do uso da linguagem Assembly e linguagens de alto nível, os métodos de preparação de programas, a sintaxe da linguagem Assembly do Z-80 e uma introdução ao seu conjunto de instruções. Fornecemos também ao leitor nesta lição o *Micro-Poster*, que será de grande utilidade para o restante do curso.

VANTAGENS DA LINGUAGEM ASSEMBLY

O programa Assembler torna mais eficiente a tarefa de programação, pois fica mais fácil a localização e correção de erros. Além disso, o Assembler indica erros de sintaxe deixando ao programador somente a tarefa de detectar erros de lógica.

Na linguagem Assembly cada instrução ou *statement* ocupa uma linha, e a cada instrução do programa fonte corresponde uma única instrução do programa objeto. Assim, a linguagem Assembly é intrínseca à arquitetura do microprocessador, o que requer o seu conhecimento pelo programador.

Além da linguagem Assembly existe também o recurso das linguagens de alto nível, tais como: Fortran, Basic, Pascal etc.

Um *statement* de uma linguagem de alto nível é convertido em uma ou mais instruções do programa objeto. Os programas que realizam tal conversão são chamados de *compiladores.*

As linguagens de alto nível têm a vantagem de não requerer do programador um conhecimento detalhado da arquitetura do processador e de acelerar o tempo de desenvolvimento dos programas. No entanto, só é possível saber o número de instruções do programa objeto após a compilação do programa.

Por outro lado, a programação em linguagem Assembly permite o controle do número de instruções e o tempo de execução do programa objeto antes da sua montagem. Mas, exige programadores habilidosos e com conhecimento da arquitetura do processador.

Portanto, a escolha entre o uso da linguagem Assembly ou de linguagens de alto nível é uma decisão de compromisso. Na maioria das aplicações em tempo real é imprescindível o uso da linguagem Assembly.

Um programa montador se classifica quanto a máquina que o executa em *residente* e *cross-assembler.* O montador é residente quando for executado no próprio microcomputador e cross-assembler quando a sua execução se faz em outro computador.

MÉTODOS DE PREPARAÇÃO DE PROGRAMAS

O programa que auxilia na preparação de um programa fonte chama-se *programa editor de textos.* O arquivo gerado pelo editor de textos é o programa fonte que, por sua vez, é a entrada do programa montador.

O programa editor de textos permite ao seu usuário manipular caracteres, palavras, linhas ou grupos de linhas do texto. A figura 1 mostra o método usual de preparação de um programa.

ERRATA

Na sétima lição, página 35, os procedimentos N = 1 e N = 2 estão trocados.

Na oitava lição, página 35, descrição da instrução PUSH, onde está escrito *"PUSH B"* leia-se *"PUSH BC"*. Idem no título da figura 15 "INSTRUÇÃO PUSH BC".

Também na oitava lição, página 36, descrição da instrução POP, onde está escrito *"POP H"* leia-se *"POP HL"*. Idem no título da figura 16, "INSTRUÇÃO POP HL".

Na mesma lição, página 32, figura 5, bloco correspondente a ULA, onde está escrito ("r") leia-se ("e"). Página 33, figura 6, faltou o sinal de ≡ (idêntico); leia-se "código de operação ≡ código da instrução". Na figura 7, página 33, bloco de operando, onde está escrito "A_1 (nn)" leia-se "A, (nn)". Na figura 10, onde está escrito "código da operação = Z8H" leia-se "código da operação = 28H". Na figura 12, página 35, bloco operando, onde está escrito "(IY + 50H) 44H" leia-se "(IY + 50H), 44H". Finalmente, na figura 13, página 35, onde está escrito "LINGUAGEM ASSEMBLER" leia-se "LINGUAGEM ASSEMBLY".

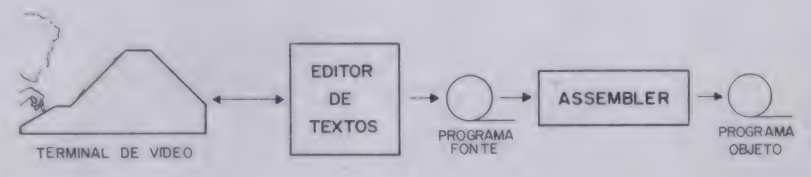


FIGURA 1 – PREPARAÇÃO DE UM PROGRAMA OBJETO

SINTAXE DA LINGUAGEM ASSEMBLY DO Z-80

A figura 2 mostra um exemplo de formato da linguagem Assembly do Z-80, assinalando os quatro campos e seus limites: endereço, código da operação, operando (s) e comentários.

O *campo de endereço* é opcional e, quando presente, pode ter até seis caracteres alfanuméricos, começando sempre por um caractere alfabético.

O *campo do código de operação* será motivo de nossas próximas lições, quando estudaremos o *conjunto de instruções.*

O *campo do(s) operando(s)* foi assunto da lição anterior, quando estudamos as *técnicas de endereçamento.*

O *campo dos comentários* é opcional e descreve de uma forma sucinta as ações das instruções. Alguns montadores delimitam os comentários através de ponto e vírgula, o que dá maior liberdade aos seus posicionamentos através do texto.

Os quatro campos formam uma linha na linguagem Assembly.

A figura 3 ilustra uma listagem típica na linguagem Assembly do Z-80, gerada pelo montador MACRO-80 da MICROSOFT.

CONJUNTO DE INSTRUÇÕES DO Z-80

Classificaremos o conjunto de instruções do Z-80, quanto a sua *funcionalidade,* nos grupos de operações abaixo:

1 – Transferências de 8 bits
2 – Transferências de 16 bits
3 – Permutas de 16 bits
4 – Transferências de blocos
5 – Pesquisa
6 – JUMP's, CALL's e RET's
7 – Lógicas e aritméticas de 8 bits
8 – Aritméticas de 16 bits
9 – Deslocamentos e rotações
10 – Manipulações e testes de bits
11 – Controles da UCP
12 – Entradas e saídas

```
                                MACRO-80 3.36   17-Mar-80       PAGE    1

                                ;********** EXEMPLO DE PROGRAMA EM ASSEMBLY Z80 **********
                                ;
0000'  21 3000                  INICIO: LD      HL,3000H        ;HL=3000H
0003'  7E                               LD      A,(HL)          ;A =(3000)
0004'  FE 00                            CP      00              ;(3000) = 00 ?
0006'  C2 000C'                         JP      NZ,PULO         ;Z=0,DESVIA
0009'  06 00                            LD      B,00            ;B=00
000B'  76                               HALT                    ;PARE
000C'  06 FF                    PULO:   LD      B,0FFH          ;B=FF
000E'  76                               HALT                    ;PARE
                                        END                     ;FIM DE MONTAGEM
        MACRO-80 3.36   17-Mar-80       PAGE    S

Macros:

Symbols:
INICIO  0000'   PULO    000C'


No Fatal error(s)
```

FIGURA 3 – LISTAGEM NA LINGUAGEM ASSEMBLY

	ENDEREÇO		CÓDIGO DA OPERAÇÃO		OPERANDO (S)		COMENTÁRIOS
COLUNAS →	1 6	7	8 11	12	13 27	28	29 132
	PROG		LD		A,B		
			CP		A,"C"		
			JP		Z,LER		
			⋮		⋮		
			⋮		⋮		
	LER		LD		(HL),A		

FIGURA 2 – UM FORMATO DE LINGUAGEM ASSEMBLY

Nas próximas lições apresentaremos as instruções do Z-80 obedecendo a classificação apresentada acima. Em alguns casos, para fins didáticos, dividiremos cada grupo de operações em subgrupos.

Neste número apresentamos um resumo do conjunto de instruções do Z-80, que denominaremos daqui para frente de *Micro-Poster.*

Os dados contidos no Micro-Poster são muito úteis à implementação de programas curtos na linguagem Assembly e que dispensam o uso do *ASSEMBLER.*

Observe que no Micro-Poster as instruções estão organizadas segundo a classificação funcional já apresentada e, além dos mnemônicos em linguagem Assembly, inclui também os seus respectivos códigos de máquina hexadecimal e os bits de condição afetados.

CONJUNTO DE INSTRUÇÕES DO MICROPROCESSADOR Z-80

A&N / Interface

MICRO-POSTER: CURSO Z-80

INSTRUÇÕES LÓGICAS E ARITMÉTICAS DE 8 BITS

	B	C	D	E	H	L	(HL)	A	n	(IX + d)	(IY + d)	S	Z	H	P/V	N	C
ADD .	8ϕ	81	82	83	84	85	86	87	C6XX	DD86XX	FD86XX	•	•	•	•	ϕ	•
ADC .	88	89	8A	8B	8C	8D	8E	8F	CEXX	DD8EXX	FD8EXX	•	•	•	•	1	•
SUB .	9ϕ	91	92	93	94	95	96	97	D6XX	DD96XX	FD96XX	•	•	•	•	1	•
SBC .	98	99	9A	9B	9C	9D	9E	9F	DEXX	DD9EXX	FD9EXX	•	•	•	•	1	•
AND .	Aϕ	A1	A2	A3	A4	A5	A6	A7	E6XX	DDA6XX	FDA6XX	•	•	1	•	ϕ	ϕ
XOR .	A8	A9	AA	AB	AC	AD	AE	AF	EEXX	DDAEXX	FDAEXX	•	•	ϕ	•	ϕ	ϕ
OR .	Bϕ	B1	B2	B3	B4	B5	B6	B7	F6XX	DDB6XX	FDB6XX	•	•	•	•	ϕ	ϕ
CP .	B8	B9	BA	BB	BC	BD	BE	BF	FEXX	DDBEXX	FDBEXX	•	•	•	•	1	•
INC .	ϕ4	ϕC	14	1C	24	2C	34	3C		DD34XX	FD34XX	•	•	•	•	ϕ	–
DEC .	ϕ5	ϕD	15	1D	25	2D	35	3D		DD35XX	FD35XX	•	•	•	•	1	–

		S	Z	H	P/V	N	C	
DAA	27	•	•	•	•	–	•	AJUSTE DE RESULTADO BCD NO AC
CPL	2F	–	–	1	–	1	–	COMPLEMENTO A UM DO AC
NEG	ED44	•	•	•	•	1	•	COMPLEMENTO A DOIS DO AC

NSTRUÇÕES ARITMÉTICAS DE 16 BITS

	BC	DE	HL	SP	IX	IY	S	Z	H	P/V	N	C
NC .	ϕ3	13	23	33	DD23	FD23	–	–	–	–	–	–
DEC .	ϕB	1B	2B	3B	DD2B	FD2B	–	–	–	–	–	–
ADD HL , .	ϕ9	19	29	39			–	–	?	–	ϕ	•
ADC HL , .	ED4A	ED5A	ED6A	ED7A			•	•	?	•	ϕ	•
SBC HL , .	ED42	ED52	ED62	ED72			•	•	?	•	1	•
ADD IX , .	DDϕ9	DD19		DD39	DD29		–	–	?	–	ϕ	•
ADD IY , .	FDϕ9	FD19		FD39		FD29	–	–	?	–	ϕ	•

INSTRUÇÕES DE DESLOCAMENTO E ROTAÇÃO

	B	C	D	E	H	L	(HL)	A	(IX + d)	(IY + d)
RR	CB18	CB19	CB1A	CB1B	CB1C	CB1D	CB1E	CB1F	DDCBXX1E	FDCBXX1E
RL	CB1ϕ	CB11	CB12	CB13	CB14	CB15	CB16	CB17	DDCBXX16	FDCBXX16
RRC	CBϕ8	CBϕ9	CBϕA	CBϕB	CBϕC	CBϕD	CBϕE	CBϕF	DDCBXXϕE	FDCBXXϕE
RLC	CBϕϕ	CBϕ1	CBϕ2	CBϕ3	CBϕ4	CBϕ5	CBϕ6	CBϕ7	DDCBXXϕ6	FDCBXXϕ6
SRA	CB28	CB29	CB2A	CB2B	CB2C	CB2D	CB2E	CB2F	DDCBXX2E	FDCBXX2E
SLA	CB2ϕ	CB21	CB22	CB23	CB24	CB25	CB26	CB27	DDCBXX26	FDCBXX26
SRL	CB38	CB39	CB3A	CB3B	CB3C	CB3D	CB3E	CB3F	DDCBXX3E	FDCBXX3E

	S	Z	H	P/V	N	C	
RR/RL	•	•	ϕ	•	ϕ	•	DESLOC. CIRC. À DIR./ESQ. ATRAVÉS DO CARRY
RRC/RLC	•	•	ϕ	•	ϕ	•	DESLOC. CIRC. À DIR./ESQ.
SRA/SLA	•	•	ϕ	•	ϕ	•	DESLOC. ARITMÉTICO À DIR./ESQ.
SRL	•	•	ϕ	•	ϕ	•	DESLOC. LÓGICO À DIR./ESQ.

		S	Z	H	P/V	N	C	
RRCA	ϕF	–	–	ϕ	–	ϕ	•	DESLOC. CIRC. DO AC À DIR.
RLCA	ϕ7	–	–	ϕ	–	ϕ	•	DESLOC. CIRC. DO AC À ESQ.
RRA	1F	–	–	ϕ	–	ϕ	•	DESLOC. CIRC. DO AC À DIR. ATRAVÉS DO CARRY
RLA	17	–	–	ϕ	–	ϕ	•	DESLOC. CIRC. DO AC À ESQ. ATRAVÉS DO CARRY
RLD	ED6F	•	•	ϕ	•	ϕ	–	DESLOC. CIRC. DE DÍGITO À ESQ. ATRAVÉS DO AC E (HL)
RRD	ED67	•	•	ϕ	•	ϕ	–	DESLOC. CIRC. DE DÍGITO À DIR. ATRAVÉS DO AC E (HL)

INSTRUÇOES DE MANIPULAÇÃO E TESTE DE BITS

		B	C	D	E	H	L	(HL)	A	(IX + d)	(IY + d)
BIT	ϕ	CB4ϕ	CB41	CB42	CB43	CB44	CB45	CB46	CB47	DDCBXX46	FDCBXX46
BIT	1	CB48	CB49	CB4A	CB4B	CB4C	CB4D	CB4E	CB4F	DDCBXX4E	FDCBXX4E
BIT	2	CB5ϕ	CB51	CB52	CB53	CB54	CB55	CB56	CB57	DDCBXX56	FDCBXX56
BIT	3	CB58	CB59	CB5A	CB5B	CB5C	CB5D	CB5E	CB5F	DDCBXX5E	FDCBXX5E
BIT	4	CB6ϕ	CB61	CB62	CB63	CB64	CB65	CB66	CB67	DDCBXX66	FDCBXX66
BIT	5	CB68	CB69	CB6A	CB6B	CB6C	CB6D	CB6E	CB6F	DDCBXX6E	FDCBXX6E
BIT	6	CB7ϕ	CB71	CB72	CB73	CB74	CB75	CB76	CB77	DDCBXX76	FDCBXX76
BIT	7	CB78	CB79	CB7A	CB7B	CB7C	CB7D	CB7E	CB7F	DDCBXX7E	FDCBXX7E
RES	ϕ	CB8ϕ	CB81	CB82	CB83	CB84	CB85	CB86	CB87	DDCBXX86	FDCBXX86
RES	1	CB88	CB89	CB8A	CB8B	CB8C	CB8D	CB8E	CB8F	DDCBXX8E	FDCBXX8E
RES	2	CB9ϕ	CB91	CB92	CB93	CB94	CB95	CB96	CB97	DDCBXX96	FDCBXX96
RES	3	CB98	CB99	CB9A	CB9B	CB9C	CB9D	CB9E	CB9F	DDCBXX9E	FDCBXX9E
RES	4	CBAϕ	CBA1	CBA2	CBA3	CBA4	CBA5	CBA6	CBA7	DDCBXXA6	FDCBXXA6
RES	5	CBA8	CBA9	CBAA	CBAB	CBAC	CBAD	CBAE	CBAF	DDCBXXAE	FDCBXXAE
RES	6	CBBϕ	CBB1	CBB2	CBB3	CBB4	CBB5	CBB6	CBB7	DDCBXXB6	FDCBXXB6
RES	7	CBB8	CBB9	CBBA	CBBB	CBBC	CBBD	CBBE	CBBF	DDCBXXBE	FDCBXXBE
SET	ϕ	CBCϕ	CBC1	CBC2	CBC3	CBC4	CBC5	CBC6	CBC7	DDCBXXC6	FDCBXXC6
SET	1	CBC8	CBC9	CBCA	CBCB	CBCC	CBCD	CBCE	CBCF	DDCBXXCE	FDCBXXCE
SET	2	CBDϕ	CBD1	CBD2	CBD3	CBD4	CBD5	CBD6	CBD7	DDCBXXD6	FDCBXXD6
SET	3	CBD8	CBD9	CBDA	CBDB	CBDC	CBDD	CBDE	CBDF	DDCBXXDE	FDCBXXDE
SET	4	CBEϕ	CBE1	CBE2	CBE3	CBE4	CBE5	CBE6	CBE7	DDCBXXE6	FDCBXXE6
SET	5	CBE8	CBE9	CBEA	CBEB	CBEC	CBED	CBEE	CBEF	DDCBXXEE	FDCBXXEE
SET	6	CBFϕ	CBF1	CBF2	CBF3	CBF4	CBF5	CBF6	CBF7	DDCBXXF6	FDCBXXF6
SET	7	CBF8	CBF9	CBFA	CBFB	CBFC	CBFD	CBFE	CBFF	DDCBXXFE	FDCBXXFE

FLAGS INFLUENCIADOS:

	S	Z	H	P/V	N	C
BIT	?	•	1	?	ϕ	–
SET	–	–	–	–	–	–
RES	–	–	–	–	–	–

NOTAS ADICIONAIS SOBRE O REGISTRO F:

Bit	7	6	5	4	3	2	1	ϕ
	S	Z	X	H	X	P/V	N	C

		SETADO	RESETADO	SETADO SE
C	CARRY-BIT	C	NC	"VAI UM" DO BIT 7
N	BIT SOMA/SUBTRAÇÃO			OPERAÇÃO DE SUBTRAÇÃO
P/V	BIT PARIDADE OVERFLOW	PE	PO	PARIDADE PAR/OVERFLOW
H	BIT HALF CARRY			"VAI UM" DO BIT 3
Z	BIT ZERO	Z	NZ	RESULTADO ZERO
S	BIT SINAL	M	P	RESULTADO NEGATIVO
X	NÃO USADO			

OPERAÇÕES DOS FLAGS:

- 1 SETADO
- ϕ RESETADO
- • DEPENDE DO RESULTADO DA OPERAÇÃO
- – NÃO AFETADO
- ? INDETERMINADO

MICRO-POSTER: CURSO Z-80

INSTRUÇÕES DE TRANSFERÊNCIA DE 8 BITS

	A	B	C	D	E	H	L	(HL)	(BC)	(DE)	(nn)	n
LD A, .	7F	78	79	7A	7B	7C	7D	7E	ϕA	1A	3AXXXX	3EXX
LD B, .	47	4ϕ	41	42	43	44	45	46				ϕ6XX
LD C, .	4F	48	49	4A	4B	4C	4D	4E				ϕEXX
LD D, .	57	5ϕ	51	52	53	54	55	56				16XX
LD E, .	5F	58	59	5A	5B	5C	5D	5E				1EXX
LD H, .	67	6ϕ	61	62	63	64	65	66				26XX
LD L, .	6F	68	69	6A	6B	6C	6D	6E				2EXX
LD (HL), .	77	7ϕ	71	72	73	74	75					36XX
LD (BC), .	ϕ2											
LD (DE), .	12											
LD (nn), .	32XXXX											

	A	B	C	D	E	H	L
LD . , (IX + d)	DD7EXX	DD46XX	DD4EXX	DD56XX	DD5EXX	DD66XX	DD6EXX
LD . , (IY + d)	FD7EXX	FD46XX	FD4EXX	FD56XX	FD5EXX	FD66XX	FD6EXX
LD (IX + d), .	DD77XX	DD7ϕXX	DD71XX	DD72XX	DD73XX	DD74XX	DD75XX
LD (IY + d), .	FD77XX	FD7ϕXX	FD71XX	FD72XX	FD73XX	FD74XX	FD75XX
LD (IX + d), n		DD36XXXX		LD (IY + d), n		FD36XXXX	

		S	Z	H	P/V	N	C
LD A, I	ED57	*	*	ϕ	*	ϕ	–
LD A, R	ED5F	*	*	ϕ	*	ϕ	–
LD I, A	ED47	–	–	–	–	–	–
LD R, A	ED4F	–	–	–	–	–	–

INSTRUÇÕES DE TRANSFERÊNCIA E PERMUTA DE 16 BITS

	BC	DE	HL	SP	IX	IY
LD . . , nn	ϕ1XXXX	11XXXX	21XXXX	31XXXX	DD21XXXX	FD21XXXX
LD . . , (nn)	ED4BXXXX	ED5BXXXX	2AXXXX	ED7BXXXX	DD2AXXXX	FD2AXXXX
LD (nn), . .	ED43XXXX	ED53XXXX	22XXXX	ED73XXXX	DD22XXXX	FD22XXXX
LD SP, . .			F9		DDF9	FDF9

	BC	DE	HL	AF	IX	IY
PUSH . .	C5	D5	E5	F5	DDE5	FDE5
POP . .	C1	D1	E1	F1	DDE1	FDE1

EX (SP), HL	E3	EX DE, HL	EB
EX (SP), IX	DDE3	EX AF, AF'	ϕ8
EX (SP), IY	FDE3	EXX	D9

BC-BC' DE-DE' HL-HL'

INSTRUÇÕES DE TRANSFERÊNCIA DE BLOCOS E PESQUISA

		S	Z	H	P/V	N	C	
LDI	EDAϕ	–	–	ϕ	*	ϕ	–	LD (DE), (HL); INC HL; INC DE; DEC BC
LDIR	EDBϕ	–	–	ϕ	ϕ	ϕ	–	COMO LDI, REPETIR ATÉ BC = ϕ
LDD	EDA8	–	–	ϕ	*	ϕ	–	LD (DE), (HL); DEC HL; DEC DE; DEC BC
LDDR	EDB8	–	–	ϕ	ϕ	ϕ	–	COMO LDD, REPETIR ATÉ BC = ϕ
CPI	EDA1	*	*	*	*	1	–	CP (HL); INC HL; DEC BC
CPIR	EDB1	*	*	*	*	1	–	COMO CPI, REPETIR ATÉ BC = ϕ ou A = (HL)
CPD	EDA9	*	*	*	*	1	–	CP (HL); DEC HL; DEC BC
CPDR	EDB9	*	*	*	*	1	–	COMO CPD, REPETIR ATÉ BC = ϕ ou A = (HL)

"JUMP'S", "CALL'S" e "RETURN'S"

	Z	NZ	C	NC	PE	PO	M	P
JP	CAXXXX	C2XXXX	DAXXXX	D2XXXX	EAXXXX	E2XXXX	FAXXXX	F2XXXX
CALL	CCXXXX	C4XXXX	DCXXXX	D4XXXX	ECXXXX	E4XXXX	FCXXXX	F4XXXX
RET	C8	Cϕ	D8	Dϕ	E8	Eϕ	F8	Fϕ
JR	28XX	2ϕXX	38XX	3ϕXX				

	INCONDICIONAIS	(HL)	(IX)	(IY)
JP	C3XXXX	E9	DDE9	FDE9
CALL	CD XXXX			
RET	C9			
JR	18XX			

RST

ϕϕ	ϕ8	1ϕ	18	2ϕ	28	3ϕ	38
C7	CF	D7	DF	E7	EF	F7	FF

DJNZ	1ϕXX	DEC B; JR NZ, d
RETI	ED4D	RETORNO DE INTERRUPÇÃO
RETN	ED45	RETORNO DE INTERRUPÇÃO NÃO MASCARADA

INSTRUÇÕES DE CONTROLE

		S	Z	H	P/V	N	C	
NOP	ϕϕ	–	–	–	–	–	–	INSTRUÇÃO DESTITUÍDA DE OPERAÇÃO
HALT	76	–	–	–		–	–	
CCF	3F	–	–	?	–	ϕ	*	COMPLEMENTA O BIT CARRY
SCF	37	–	–	ϕ	–	ϕ	1	SETA O BIT CARRY
EI	FB	–	–	–	–	–	–	HABILITA INTERRUPÇÃO
DI	F3	–	–	–	–	–	–	INIBE INTERRUPÇÃO
IM ϕ	ED46	–	–	–	–	–	–	INTERRUPÇÃO MODO - 0
IM 1	ED56	–	–	–	–	–	–	INTERRUPÇÃO MODO - 1
IM 2	ED5E	–	–	–	–	–	–	INTERRUPÇÃO MODO - 2

INSTRUÇÕES DE ENTRADA E SAÍDA

	A	B	C	D	E	H	L	S	Z	H	P/V	N	C
IN . , (C)	ED78	ED4ϕ	ED48	ED5ϕ	ED58	ED6ϕ	ED68	*	*	*	*	ϕ	–
OUT (C), .	ED79	ED41	ED49	ED51	ED59	ED61	ED69	–	–	–	–	–	–

		S	Z	H	P/V	N	C	
IN A,(n)	DBXX	–	–	–	–	–	–	
OUT(n), A	D3XX	–	–	–	–	–	–	
INI	EDA2	?	*	?	?	1	–	IN (HL), (C); INC HL; DEC B
INIR	EDB2	?	1	?	?	1	–	COMO INI, REPETIR ATÉ B = ϕ
IND	EDAA	?	*	?	?	1	–	IN (HL), (C); DEC HL; DEC B
INDR	EDBA	?	1	?	?	1	–	COMO IND, REPETIR ATÉ B = ϕ
OUTI	EDA3	?	*	?	?	1	–	OUT (C), (HL); INC HL; DEC B
OTIR	EDB3	?	1	?	?	1	–	COMO OUTI, REPETIR ATÉ B = ϕ
OUTD	EDAB	?	*	?	?	1	–	OUT (C), (HL); DEC HL; DEC B
OTDR	EDBB	?	1	?	?	1	–	COMO OUTD, REPETIR ATÉ B = ϕ

CALENDÁRIO

INTERFACE PROMOVE DOIS SEMINÁRIOS NO RIO DE JANEIRO

A revista INTERFACE, com o apoio do grupo de micro da Telerj e do projeto CIRANDA da Embratel, irá realizar nos próximos dias 3, 10, 17 e 24 de setembro, no Clube de Engenharia do Rio de Janeiro, o **I ENCONTRO REGIONAL DOS USUÁRIOS DE MICROCOMPUTADORES,** onde serão debatidos os problemas enfrentados pelo usuário de microcomputador como: manutenção, incompatibilidade de software, apoio do fabricante, troca de programas, novas tecnologias, CPM e grupos de micros.

Também no mês de setembro, dias 16 e 17, realizará o **I SEMINÁRIO SOBRE APLICAÇÃO NO MICROCOMPUTADOR NA ÁREA MÉDICA,** com o apoio da Sociedade de Medicina e Cirurgia do Rio de Janeiro, no Colégio Brasileiro de Cirurgiões.

No seminário serão apresentadas as múltiplas aplicações do microcomputador, conceitos básicos de hardware e software, características e potencialidades do microcomputador, aplicações mais comuns e específicas, e onde melhor utilizá-lo no consultório, clínica e hospital.

Informações e inscrições na Sociedade de Medicina e Cirurgia do Rio de Janeiro, Av. Men de Sá, 197 — Tel.: (021) 232-2888 e no CBI — Centro Brasileiro de Informática, Av. Passos, 115 — sala 215 — Tel.: (021) 233-1123.

CURSOS/CEPUERJ

O Instituto de Matemática e Estatística da UERJ, em ação conjunta com o CEPUERJ (Centro de Produção Universidade do Estado do Rio de Janeiro), vem desenvolvendo atividades na área de informática visando o desenvolvimento de recursos humanos e o atendimento da comunidade em geral.

São os seguintes os cursos programados para o 2º semestre:

• *Tecnologia de Programação de Computadores em Linguagem Cobol*

Objetiva desenvolver habilidades referentes à tecnologia de programação de computadores eletrônicos digitais utilizando a linguagem COBOL, tendo em vista o desenvolvimento de programas eficientes para sistemas automáticos de tratamento de informações comerciais e administrativos de pequeno, médio e grande portes.

O período de realização é de 22 de agosto a 28 de novembro e a carga horária é de 120 horas.

Horário: 2ªs, 4ªs e 6ªs feiras, das 19:00 às 22:00 horas.

• *Análise e Projeto Estruturado de Sistemas de Informação*

Seu objetivo é apresentar os métodos estruturados de análise, especificação e projetos de sistemas de informação, tendo em vista fornecer uma base metodológica para:

— abordagem sistêmica das funções da empresa, considerando aspectos culturais do ambiente.

— estudar os caminhos da madança, a partir de uma visão geral, objetivando a implementação de normas, procedimentos e sistemas automatizados ou não.

A sua carga horária é de 90 horas e se realizará de 22 de agosto a 27 de outubro às 2ªs, 3ªs e 5ªs feiras, das 19:00 às 22:00 horas.

• *Aplicação dos Computadores Eletrônicos na Engenharia*

Seu objetivo é desenvolver habilidades referentes à tecnologia de utilização do processamento eletrônico de dados nas funções de engenharia.

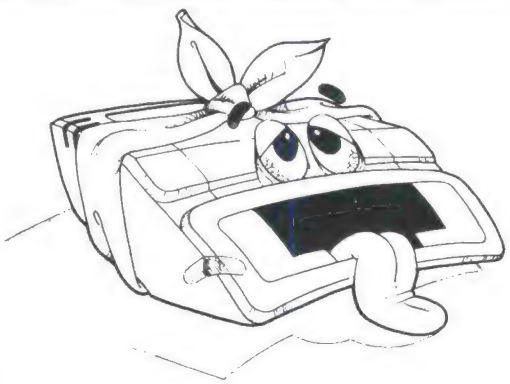
O curso se realizará de 12 de setembro a 11 de outubro nas 2ªs, 3ªs, 4ªs e 5ªs feiras, das 19:00 às 22:00 horas.

SEMINÁRIO SOBRE APLICAÇÃO DE MICROS

A J. R. de Góes Com. e Assessoria, empresa atuante no mercado de Formulários Contínuos e de Microcomputadores, promoverá no mês de setembro, em consórcio com outras empresas do ramo, um Seminário intitulado **O Micro no Lar e nos Negócios,** visando intensificar as aplicações mais sérias, tanto no campo pessoal quanto no empresarial, no intuito de que os usuários e interessados em informática obtenham mais rápida e eficientemente os resultados positivos na relação custo/desempenho, desmistificando assim o mistério dos microcomputadores.

O Seminário, além de convidados representativos da área, constará de palestras, debates, cases studies, projeção de filmes e slides, cursos paralelos, demonstrações e treinamentos em equipamentos. Ao término, será entregue o certificado de participação e, se o Leão autorizar, haverá sorteio de um micro aos participantes.

Maiores informações através do telefone 234-3945, Rio de Janeiro.



apII unitron

Comprado na MICROEQUIPO, nunca é abandonado. Mantemos equipe técnica treinada na fábrica para que seu computador receba a melhor assistência em hardware ou software.

Consultem-nos. Temos os melhores preços e condições para você ter o seu computador apII unitron.

MICROEQUIPO
Comércio, Representações e Serviços Ltda.
Rua Álvaro Alvim, 37 - Grupo 519
20031 - Rio de Janeiro - RJ
Tel. (021) 220-5059

Interface

CURSO DE PROGRAMAÇÃO EM LINGUAGEM BASIC

1ª Lição – *Por José Arthur da Rocha*

Este curso tem como principal objetivo capacitar o leitor no desenvolvimento e na execução de programas próprios, e na análise e otimização de programas escritos na linguagem BASIC.

Para atingir este objetivo serão vistas técnicas de programação estruturada e sua implementação utilizando a linguagem BASIC, ao longo das doze lições que compõem este curso. Ele já foi aplicado diversas vezes pelo autor e para um público sempre heterogêneo, mostrando uma boa absorção pelos alunos, sem que eles tivessem quaisquer conhecimentos anteriores no assunto. Portanto, é dirigido àqueles que desejam aprender a "conversar" com um computador, tirando e colocando nele informações necessárias ao cumprimento de seus fins específicos, pouco importando a área de aplicação destes fins e o nível intelectual daquelas pessoas.

Como é próprio de qualquer curso e principalmente nos de programação, o trabalho prático é essencial. Portanto, os leitores que tiverem a disponibilidade de um microcomputador para praticarem, certamente terão um melhor aproveitamento. Como trabalho prático, serão propostos ao final de cada lição pequenos problemas cuja solução será dada na lição seguinte. É claro que o leitor que, além destes problemas propostos, procurar outros para se exercitar atingirá mais rapidamente o seu objetivo.

Aqui mostraremos a constituição de um computador e como interagem seus diferentes blocos. Veremos que para realizar uma determinada tarefa ela será dividida em subtarefas que devem ser cuidadosamente definidas, para que o processo não perca a sua lógica. Para chegar a alcançar este objetivo nosso processo passará por uma série de otimizações até atingirmos o chamado PROGRAMA.

CONSTITUIÇÃO DO COMPUTADOR

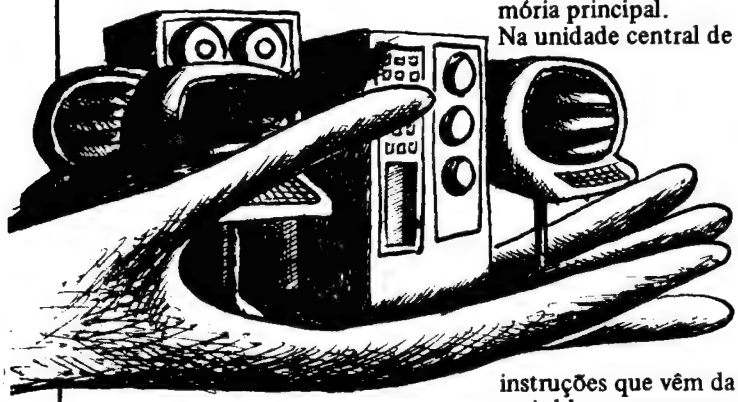
Podemos definir o computador como uma máquina capaz de receber dados, operá-los matematicamente e guardar ou fornecer ao operador o resultado destas operações. É uma definição bem resumida tendo em vista as máquinas que hoje conhecemos. No entanto, ela pode ser estendida a todas as máquinas capazes de realizar aquelas funções, desde a primeira que se tem notícia: o ábaco. Podemos encontrar um histórico do computador nas referências 1 e 2.

Todos os computadores, desde as máquinas mais complexas até os mais simples microsistemas, são constituídos por três blocos principais: dispositivos de entrada e saída, memória e unidade central de processamento.

Os dispositivos de entrada e saída são os responsáveis pela comunicação do computador com o meio externo. Os dispositivos de entrada permitem que os dados (ou instruções, como veremos mais tarde) possam ser transmitidos ao computador. Os dispositivos de entrada mais comuns são o teclado, leitora de cartões, terminais de vídeo e teleimpressoras. Os dispositivos de saída permitem a obtenção dos resultados fornecidos pelo computador. Os dispositivos de saída mais comuns são as impressoras e uma vez mais os terminais de vídeo e as teleimpressoras, que servem ambos para entrada e saída de dados.

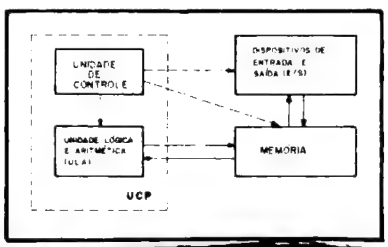
O bloco memória pode ser dividido em duas partes. Uma delas, residente no computador e nor-

malmente constituída por circuitos eletrônicos integrados, é chamada memória principal. A outra, não residente, é chamada memória secundária ou externa (fita magnética e disquete são exemplos). As duas formas de armazenamento têm características e propósitos diferentes. Programas em execução e dados necessários a esta execução residem na memória principal. Informações não imediatamente necessárias são normalmente colocadas na memória secundária, cuja capacidade é quase ilimitada, embora o tempo de acesso para a recuperação da informação seja grande em relação as da memória principal.



Na unidade central de processamento (UCP) é executada grande parte das tarefas realizadas pelo computador e é de onde provêm todos os sinais que ordenam a realização das mesmas. Estas são as características dos dois blocos que a compõe: unidade lógica e aritmética (ULA) e unidade de controle. A ULA é responsável por todos os cálculos que envolvem operações aritméticas como adição, subtração, multiplicação e divisão, e operações lógicas como comparação entre dois valores e funções booleanas. A unidade de controle, o mais importante de todos os blocos, é responsável pela decodificação e execução de instruções que vêm da memória, enviando sinais de controle aos demais blocos para que realizem suas funções, cada um a seu tempo, na execução de um programa.
Todos estes blocos estão representados na figura 1.

DIAGRAMA DE BLOCOS DE UM COMPUTADOR

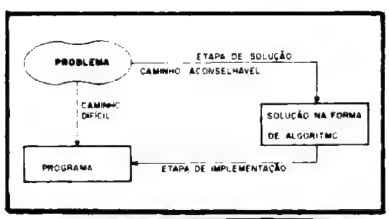


Os blocos apresentados acima compõem o chamado hardware do computador. Para o acionamento deste hardware é necessário um PROGRAMA, que é uma seqüência de instruções dadas à UCP para a realização de alguma tarefa. Este programa deve ser expresso numa forma que possa ser entendida pela UCP (linguagem de máquina). Esta é uma tarefa difícil já que cada instrução deve ser codificada pelo operador antes de ser introduzida no computador. No entanto, existem programas que compõem o chamado SOFTWARE do computador, que "traduzem" para ele a linguagem utilizada pelo programador. Esta linguagem é normalmente de fácil compreensão e entre elas estão o FORTRAN, o COBOL e o BASIC. Mais informações sobre o SOFTWARE do computador e sua relação com o HARDWARE podem ser encontradas nas referências 2 e 3.

ALGORITMOS

O computador não faz necessariamente o que nós queremos que ele faça e sim o que nós mandamos ele fazer. Em outras palavras, a nossa idéia e a maneira com que desejamos que ela seja executada devem ser colocadas claramente para o computador, não podendo haver ambigüidade nas instruções do nosso programa. O computador sempre toma um curso na execução de uma tarefa. Devemos ter o cuidado de nos assegurarmos que somente um único curso possível deve existir para ser atingido o resultado esperado, fornecendo o máximo de informações para que seja atingido este fim. Por exemplo, a declaração "calcule a média dos resultados de um teste" é muito imprecisa para ser executada. Falta ser especificado, por exemplo, quais são os resultados e quantos eles são.
Portanto, é importante dividirmos a tarefa a ser executada em duas etapas: solução e implementação. Na etapa "solução" deverá ser projetado um ALGORITMO para a solução do problema. Um algoritmo pode ser definido como uma seqüência de passos para a solução de um problema. Feito o algoritmo, estando seguros que neste não há informações ambíguas que possibilitem interpretações altenativas, passamos à etapa de "implementação" deste algoritmo em alguma linguagem de programação (o BASIC, por exemplo). Um algoritmo suficientemente preciso e detalhado proporciona uma passagem quase que imediata para o programa. A execução de uma tarefa está representada na figura 2.

SOLUÇÃO DE UM PROBLEMA



Para ilustrar as considerações acima vamos projetar um algoritmo para um processo familiar a todos: trocar o pneu de um automóvel. A operação pode ser expressa em apenas dois passos:

1. retirar o pneu furado;
2. colocar o estepe.

Estes parecem resolver o problema, no entanto não estão adequados aos nossos propósitos. O primeiro passo, "retirar o pneu furado", deve ser mais especificado (há pessoas que não sabem como retirar o pneu). Assim, ele pode ser expandido em: • *Levante o carro com o macaco.* • *Desaperte as porcas.* • *Retire o pneu furado.* O segundo passo, da mesma forma, poderia ser expandido em: • *Coloque o estepe.* • *Aperte as porcas.* • *Abaixe o carro.*

Não atingimos ainda a especificação necessária. Alguns dos passos deste algoritmo necessitam um melhor detalhamento para que sejam executados. Por exemplo, o passo "desapertar as porcas" mostra uma ação que deve ser repetida ("desapertar uma porca") até que todas tenham sido retiradas. Isto requer as seguintes operações: – *desaperte uma porca; – se ainda faltam porcas a serem retiradas volte ao passo anterior, se não continue.* Deve-se ter o mesmo procedimento para o passo "apertar as porcas". Quem já trocou pneu de carro sabe que um teste deve ser feito antes de se tomar qualquer uma das providências acima: verificar o estado do estepe. Este seria mais um passo para o nosso algoritmo, talvez o primeiro a ser executado:

• *Se o estepe estiver em mau estado, chame o socorro e FIM; do contrário, continue.* Após todas essas alterações, nosso algorítmo agora é:

1. se o estepe estiver em mau estado, chame o socorro e FIM; do contrário, continue.
2. desaperte uma porca;
3. se ainda faltam porcas a serem retiradas volte ao passo anterior; se não continue;
4. retire o pneu furado;
5. coloque o estepe;
6. aperte uma porca;
7. se faltam porcas a serem apertadas volte ao item anterior; se não continue;
8. abaixe o carro.

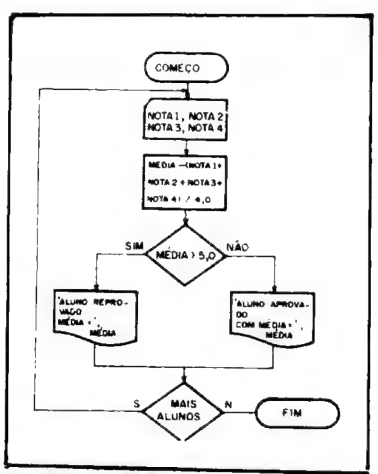
Como podemos observar, este processo de detalhamento pode seguir indefinidamente. No entanto, o exemplo apresentado serve como ilustração de como se deve projetar um algoritmo. As operações feitas são simples e não duvidosas. O método com que ele foi desenvolvido é importante: começamos com uma solução geral do problema e chegamos ao final do algoritmo pelo aumento sistemático do nível de detalhamento. Como podemos saber qual é o nível suficiente de detalhamento? Isto é função do agente que eventualmente executará o algoritmo. Os computadores, por exemplo, têm um conjunto de instruções limitado e o algoritmo, em sua forma final, deve ser expresso segundo este conjunto.

Os algoritmos apresentados estão na forma chamada NARRATIVA, onde os passos são colocados em linguagem natural, que nem sempre é um bom veículo de transferência de informação: a compreensão da informação pode sacrificar a precisão. Geralmente, a representação gráfica pode substituir a verbal na colocação de uma idéia. No caso de algoritmo podemos dizer que vem a calhar, principalmente no que diz respeito à transferência de informação. A forma gráfica de representação de um algoritmo chama-se FLUXOGRAMA. O fluxograma mostra a lógica de um algoritmo, enfatizando cada passo e suas interconexões. Existem alguns símbolos que foram padronizados e são mostrados na figura 3.

SÍMBOLOS MAIS UTILIZADOS EM FLUXOGRAMAS

"Atribuição" é a execução direta de uma tarefa (p. ex. uma operação matemática). "Decisão" testa uma ou mais condições necessárias ao andamento do algoritmo. As outras duas caixas simbolizam a comunicação do algoritmo com o meio externo. As caixas são interconectadas por linhas diretas indicando a ordem em que as operações devem ser realizadas. Um exemplo simples de fluxograma é mostrado na figura 4, onde é calculada a aprovação de alunos através da média obtida em provas.

FLUXOGRAMA PARA CÁLCULO DA MÉDIA DE 4 NOTAS



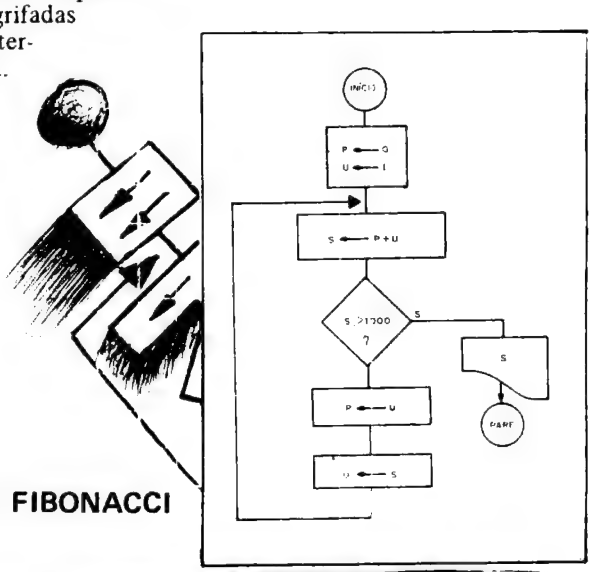
No exemplo numérico de fluxograma da figura 4 foram utilizados nomes para identificar cada uma das notas e a média das quatro notas, que são valores numéricos. Estes "nomes" são chamados VARIÁVEIS. Na caixa de entrada de dados vemos que as quatro notas que entram no algoritmo são atribuídas, respectivamente, as variáveis NOTA 1, NOTA 2, NOTA 3 e NOTA 4. Na caixa seguinte é feita uma atribuição a uma variável, isto é, é dado à variável MÉDIA o resultado da expressão composta pelas quatro variáveis anteriores. Nas caixas de saída de dados aparecem mensagens e novamente a variável MÉDIA. Aí serão postos à disposição do usuário o valor da média obtida na caixa de atribuição e o resultado.

do aluno, que obviamente depende desta média. O algoritmo acima está detalhado suficientemente para ser implementado em qualquer linguagem, formando o programa.

Um exemplo completo

Com base no que foi até agora visto nesta lição façamos um algoritmo que determine o primeiro número maior que 1000 na seqüência de Fibonacci.
Primeiro precisamos saber a lei de formação desta seqüência. Os dois primeiros termos da seqüência de Fibonacci são 0 e 1. Os termos a seguir são a soma dos dois precedentes. Portanto, os primeiros termos da seqüência de Fibonacci são: 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, . .
O problema pede, então, que calculemos o primeiro termo maior que 1000 desta seqüência.
Pela própria definição da lei de formação, sabemos que o próximo termo da seqüência será a SOMA do ÚLTIMO termo com o PENÚLTIMO. As três palavras grifadas anteriormente serão utilizadas como variáveis em nosso algoritmo. Este terminará quando a variável SOMA for maior que 1000 e deverá imprimi-la.
Podemos descrever esta ação com o seguinte algoritmo:

1. Dê valores iniciais a ULTIMO (0) e a PENÚLTIMO (1);
2. Ache a soma do ÚLTIMO com PENÚLTIMO;
3. O valor SOMA é maior que 1000?
4. Se for, escreva o valor SOMA e pare;
5. Se não for, passe a chamar ÚLTIMO de PENÚLTIMO;
6. Passe a chamar SOMA de ÚLTIMO;
7. Volte para o passo 2.



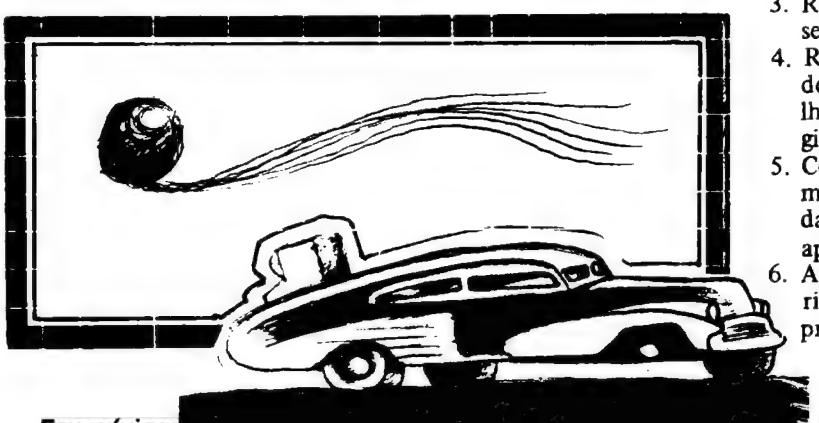
Utilizando as variáveis P, U e S para representar, respectivamente, o penúltimo e último termo e a soma deles, é apresentado na figura 5 o fluxograma-solução do problema.

FLUXOGRAMA PARA O PROBLEMA DA SEQÜÊNCIA DE FIBONACCI

SUMÁRIO

Nesta lição tentamos dar ao leitor uma idéia de como é a organização de um computador típico e como ele opera. Poderíamos ter explorado estas áreas mais detalhadamente, mas isto fugiria ao escopo de nosso curso.

Para solução de problemas computacionais, sugerimos o seguimento dos seguintes passos:



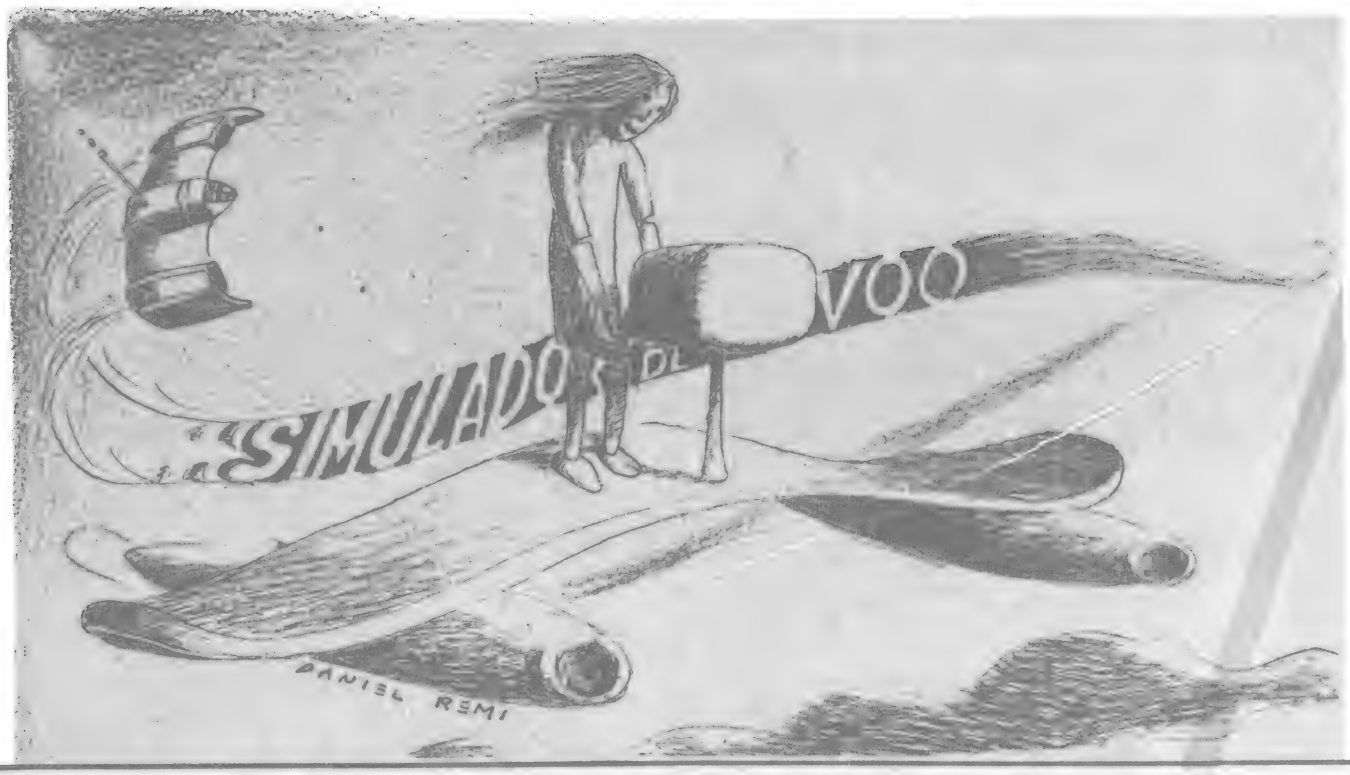
1. Entender as especificações do problema proposto. Uma maneira prática de realizar isto é prever as "saídas" do problema para várias "entradas".
2. Apresentar um algoritmo geral para a solução do problema, ficando atento a detalhes específicos.
3. Relacionar todas as variáveis que você ache que serão necessárias, indicando o propósito delas.
4. Retornar aos primeiros passos do algoritmo e detalhá-los. Esteja certo de que os passos detalhados cumprem a mesma função do passo original.
5. Concluindo que o algoritmo já está suficientemente detalhado, novamente coloque as "entradas" do passo 1 e observe se as saídas esperadas aparecem no final do algoritmo.
6. Agora você está apto a implementar o algoritmo em uma linguagem de programação apropriada: o BASIC, por exemplo.

Exercícios

1) Usando o método descrito nesta lição, desenvolva um algoritmo que resolva o seguinte problema: – *trocar uma lâmpada queimada no teto de uma sala, sabendo que uma similar boa está numa caixa com várias outras lâmpadas.* 2) Modificar o fluxograma da seqüência de Fibonacci para calcular: – *qual a posição dentro da seqüência do primeiro número maior que 1000? – qual é o 100º número? – qual é a soma dos 30 primeiros termos da seqüência? As respostas para estes problemas estarão na lição número 2 do nosso curso.*

Referências

1) Rubens, André G. e Oliveira, Ney A. "Histórico da Computação" Curso de Microprocessadores Z-80 p. 23 – 1982. Interface nº 2. 2) Tremblay, JP e Bunt, RB. "An Introduction to Computer Science". Computer Science Series – Mc GrawHill 1981. 3) Costa, Cesar da "Software Básico X Software de Aplicação". Interface nº 3 p. 24 1983. 4) Forsythe, Al e Keenan, T. A. "Ciência de Computadores – 1". Livro Técnico SA 1972.



Alguns programas em Basic trazem em sua primeira linha REM uma rotina em linguagem de máquina que não é apresentada em sua listagem; em seu lugar a listagem fornece um conjunto de caracteres do repertório do microcomputador. Este é o caso do programa **SIMULADOR DE VÔO,** *publicado na seção "Clube do Microsoftware", revista número 5.*

Neste artigo apresentamos os códigos hexadecimais correspondentes à linha REM; um programa que cria uma área para a rotina em assembler e controla a digitação dos 2.900 códigos; e um outro programa para armazenar a rotina em linguagem de máquina na memória do micro. Apresentamos também a listagem do **SIMULADOR DE VÔO,** *a descrição dos símbolos gráficos, as instruções de como jogá-lo e um procedimento para se obter os códigos hexadecimais da primeira linha REM de um programa, a partir do programa em fita cassete (o programa* **SIMULADOR DE VÔO** *em fita cassete pode ser obtido no Rio: LOJA MICROKIT e em São Paulo: MICROSOFT – brinde na assinatura da revista "Microhobby").*

Sugerimos aos leitores que não conseguirem rodar o programa escrever-nos relatando as dúvidas, que serão publicadas e respondidas nesta seção. "A programação em muito se parece com o jogo de xadrez, onde qualquer um pode aprender os movimentos, mas poucos tornam-se grandes mestres; aprenda e pratique os movimentos e talvez você tenha talento para seguir adiante". Mãos à obra; analise cada etapa, tente eliminar os possíveis erros apresentados, recorra aos amigos, enfim, não desista.

Colaboraram na elaboração deste artigo os leitores e usuários do microcomputador TK-82C: Engº Marcio Hampshire de Araújo e o programador Francisco José Domingues Figueiredo.

CRIANDO UMA ÁREA PARA A ROTINA EM ASSEMBLER

Vamos criar uma área para a rotina em Assembler e controlar o número de códigos digitados, digitando o seguinte programa:

```
1 REM
2 LET F = PEEK 16511 + 256 *PEEK 16512
3 IF F< 2928 THEN PRINT "FALTAM";
    2928 – F; "CARACTERES"
4 IF F > 2928 THEN PRINT "RETIRE";
    F – 2928; "CARACTERES"
5 IF F = 2928 THEN PRINT "ATINGIU O
    NÚMERO DE CARACTERES"
6 STOP
```

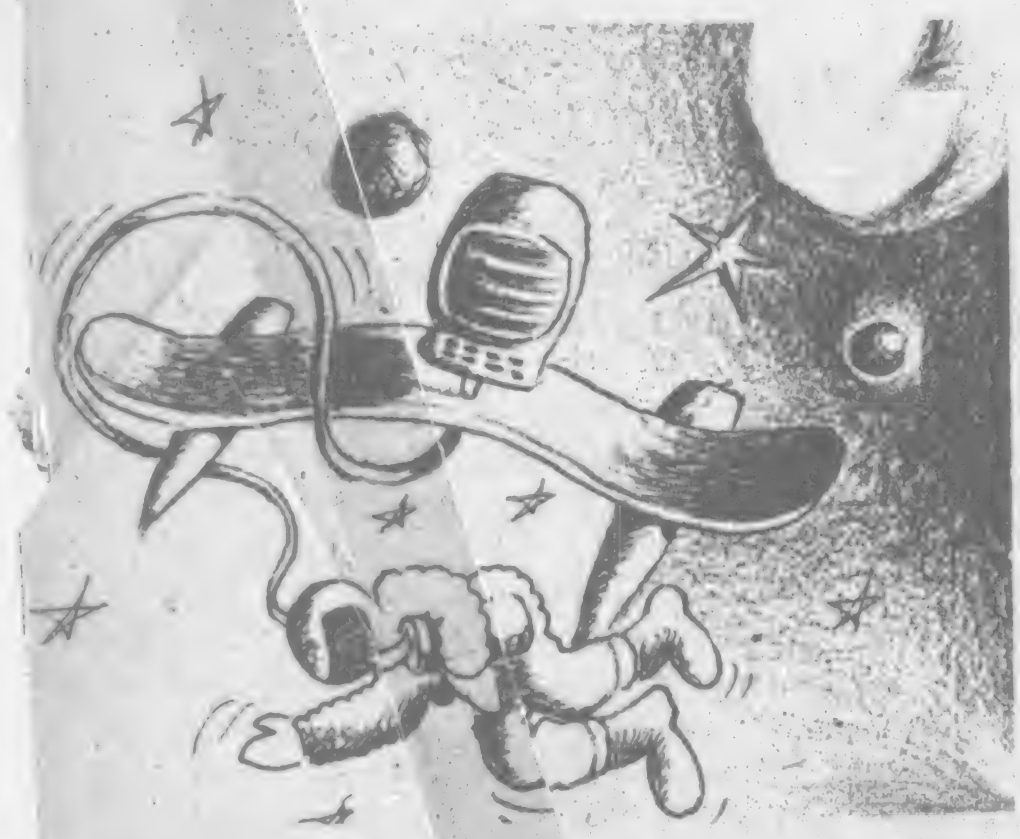
Após este procedimento edite a linha 1 (EDIT + SHIFT1) e digite os 2928 códigos hexadecimais da tabela 1. Em determinados momentos entre com o comando *GOTO 2* e verifique a quantidade de caracteres que faltam ou excedem a tabela 1. Finalmente, após a digitação dos códigos, retire as linhas 2, 3, 4, 5 e 6 do programa.

Alguns leitores deverão estranhar a quantidade de códigos da tabela 1, que é maior do que o número de caracteres apresentados na listagem da primeira linha REM (a listagem apresenta 332 caracteres que equivalem aos códigos apresentados do endereço 16514 até o endereço 16845). A listagem não apresenta o restante dos códigos.

PROGRAMA PARA ARMAZENAR A ROTINA EM LINGUAGEM DE MÁQUINA

Agora, após a entrada de todos os códigos da tabela 1, vamos entrar com um programa que armazenará a rotina em linguagem de máquina na memória do TK ou CP.

```
10 LET X = 16514
20 LET Y$ = "" (duas aspas, uma após a outra)
25 LET W = 0
30 IF Y$ = "" THEN INPUT Y$
                (duas aspas, uma após a outra)
40 IF Y$ = "P" THEN STOP
50 POKE X, 16 *CODE Y$ + CODE Y$ (2) – 476
60 SCROLL
70 PRINT AT 21, 0, X; "" ; Y$ (TO 2)
                (dois espaços entre as aspas)
80 IF W = 20 THEN INPUT M$
90 IF W = 20 THEN LET W = –1
100 LET W = W + 1
105 LET X = X + 1
110 LET Y$ = Y$ (3 TO)
120 GOTO 30
```



Execute o programa, e na tela após surgir o endereço 9441 com o seu conteúdo o programa encontrará no próximo endereço a letra "P", e parará automaticamente com código 9/40, no canto inferior esquerdo da tela. Dê um LIST e veja o conteúdo da tabela 1 na primeira linha REM; a seguir retire as linhas 10, 20, 25, 30, 40, 50, 60, 70, 80, 90, 100, 105, 110 e 120 do programa. Se você executou corretamente todos os itens anteriores até aqui, a primeira linha REM do programa *SIMULADOR DE VÔO* está armazenada na memória do seu micro. Agora entre com o restante do programa normalmente, a partir da linha 2 da listagem. Os simbolos gráficos são descritos a seguir e o número entre parênteses indica o número de vezes que a função é repetida.

Quando da execução deste programa aqui na redação, surgiram alguns problemas que poderão ocorrer com os leitores:

- Os 16 primeiros endereços (16514 a 16529) alteram o seu conteúdo a cada listagem; ignoramos.
- Erro de código 4, que eliminamos retirando a palavra *VOCE* nas linhas de PRINT ou LET do programa.

Os leitores que conseguirem melhores soluções enviem-nos para publicação.

```
  1 REM COPY COPY  ▒?▒E??▒X= ▒R
▬??▒?E?  LOAD ▪ ?▪ ?OTAN ??▒?2? LOA
D ▪ ?▪ ?OTAN ▪▒2?▒ LOAD ▬▪ OTAN ??
▪ LOAD ▪ ?OTAN NOT 2▒R LPRINT ?▒
M▒▪ )= LOAD ▪ "" ▬ STEP ▬"" ▬ AT ▬
 LPRINT ??▪ STEP ▬▪OTAN 22▬NOT L
OAD ▪ ?OTAN COPY ▒▬?▒??????? LOA
D ▪ "" ▬ STEP ▬AT ▬▪ "" ▬ STEP ??
LPRINT ▪▒▪ ?▒OTAN CC?? LOAD "" ST
EP ▪ "" STEP ▪ ?▪ "" STEP ▬▪ STEP
STEP ?▒ STEP ?OTAN WCOS ▒▪ LOAD
??▒OTAN. COPY ?LN ▒▬? LOAD ▪ ?▒?OT
AN ?ABS ▒▬? LOAD ▪ ?▒?OTAN COPY ▪
4▒▒ COPY ▒4?▪ ▒(4? ▪?(? COPY ▪ (▒?
 COPY SAVE 4 ▪M▪ ▒▒4▪ M COPY ▪ (R▪ ▒▪
▒R COPY ▪ ▪4▒▪ COPY ▒4▒▪ COPY ▒▒9
▪ COPY COPY ▪ ▪▒▒"$?>=-/,13579BD
FIKMOQSUWYZINKEY$??????????????????
????????????????????????
```

```
 2▒REM
10 DIM B$(3)
11 DIM G$(4)
12 DIM C(44)
13 DIM L$(32)
14 DIM M$(32)
15 LET Y$=""
16 LET Z$=""
17 DIM P$(116)
18 LET P$(112)="9"
19 LET P$(111)="1"
20 GOSUB 9000
25 GOSUB 4000
55 LET C(16)=VAL "5"
60 LET C(31)=VAL "12"
61 LET C(32)=VAL "16"
62 LET C(33)=VAL "2"
63 LET C(34)=515
64 LET C(35)=180/PI
65 LET C(36)=VAL "5"
66 LET C(37)=VAL "20"
67 LET C(38)=VAL "12"
68 LET C(40)=200
69 LET C(41)=VAL "1"
73 LET C(22)=9/127
75 LET C(24)=.005
76 LET C(25)=.6/(127*3600)
77 LET C(26)=.001
79 LET C(28)=VAL ".5"
80 LET P$(6)="P"
81 LET P$(12)="M"
82 LET P$(114)="2"
83 LET P$(57)=" ."
84 LET P$(34)="U"
85 LET P$(11)="A"
86 LET P$(15)=P$(6)
87 LET P$(83)="P"
88 LET P$(70)=P$(83)
102 LET C(12)=VAL "12"
103 LET C(8)=VAL "5"
105 LET C(2)=150
106 LET C(5)=VAL "45"
107 LET P$(106)="*"
108 LET P$(107)=P$(106)
109 LET P$(115)=P$(107)
110 LET P$(116)=P$(106)
111 LET P$(108)="("
112 LET P$(9)="G"
113 LET P$(20)="N"
114 LET P$(110)=")"
115 LET P$(21)="C"
116 LET P$(74)="B"
117 LET P$(1)="E"
118 LET P$(91)="."
200 GOSUB 4500
304 LET C(19)=R*SIN A
306 LET C(20)=R*COS A
352 POKE 16518,T-256*INT (T/256
353 POKE 16519,INT (T/256)
380 LET C(21)=VAL "0"
381 LET G$="REC."
999 SLOW
```

TABELA 1

Posição	Conteúdo																								
16514	FF	FF	ϕϕ	ϕ8	67	ϕ8	AA	44	70	ϕ3	BD	94	ϕϕ	Bϕ	37	ϕ3	73	73	91	69	1E	5A	EF	ϕ4	ϕF
16539	ϕ4	ϕF	34	C9	78	78	91	6E	1E	5F	EF	ϕ4	ϕF	ϕ4	ϕF	34	C9	87	8C	1E	5ϕ	AF	EF	ϕ3	ϕ4
16564	ϕ4	34	C9	23	23	64	87	EF	ϕ4	ϕF	34	C9	D7	1E	B4	37	E1	7D	B9	32	BE	87	DC	EF	ϕ4
16589	Cϕ	ϕ2	ϕ4	E0	ϕ3	Cϕ	ϕ2	ϕ4	C1	ϕ2	ϕ4	E1	ϕF	ϕF	ϕ4	E ϕ	ϕ1	ϕ3	34	C9	1E	1E	96	D7	EF
16614	ϕ4	ϕF	34	C9	FF	9B	Aϕ	C3	Aϕ	5F	78	5A	73	5A	78	5F	73	EF	ϕ4	Cϕ	ϕ2	ϕ4	Eϕ	ϕ3	C1
16639	ϕ2	ϕ4	Cϕ	ϕ2	ϕ4	Bϕ	ϕF	ϕF	E1	ϕ1	ϕ6	ϕ4	ϕF	24	34	C9	28	28	78	73	EF	Cϕ	Eϕ	ϕ4	ϕ1
16664	Cϕ	Eϕ	ϕ4	ϕF	ϕ1	Cϕ	F0	ϕ4	ϕ3	F0	Eϕ	ϕF	ϕ5	Eϕ	ϕF	34	C9	3C	C8	Aϕ	87	EF	ϕF	ϕF	24
16689	34	C9	FF	46	CD	Aϕ	82	5A	EF	ϕ4	ϕF	24	ϕF	34	C9	4B	D2	Aϕ	82	5F	EF	ϕ4	ϕF	24	ϕF
16714	34	C9	FF	ϕ1	2ϕ	A5	ϕ8	FF	ϕ8	2ϕ	55	ϕ1	ϕ8	1ϕ	2ϕ	5ϕ	ϕ1	ϕF	1ϕ	1ϕ	5ϕ	FF	ϕ1	1ϕ	ϕ8
16739	55	FF	F8	2ϕ	ϕ2	32	ϕ1	ϕA	2ϕ	24	32	FF	ϕϕ	ϕ2	1ϕ	37	ϕ1	ϕ8	ϕ2	ϕ8	37	FF	ϕϕ	ϕ2	2ϕ
16764	A5	ϕ1	FF	8ϕ	2ϕ	A5	ϕ2	FF	8ϕ	08	A5	ϕ4	FF	FF	ϕϕ	ϕ2	ϕ4	ϕ7	ϕ9	ϕB	ϕD	ϕF	12	14	16
16789	18	1A	1D	1F	21	23	25	27	29	2B	2E	3ϕ	32	34	36	38	3A	3C	3E	3F	41	43	45	47	49
16814	4B	4C	4E	5ϕ	52	53	55	57	58	5A	5B	5D	5E	6ϕ	61	63	64	65	67	68	69	6B	6C	6D	6E
16839	6F	7ϕ	71	72	73	74	75	76	77	77	78	79	79	7A	7B	7B	7C	7C	7D	7D	7D	7E	7E	7E	7F
16864	7F	7F	7F	7F	7F	7F	7F	7F	7F	7F	7E	7E	7E	7D	7D	7D	7C	7C	7B	7B	7A	79	79	78	77
16889	77	76	75	74	73	72	71	7ϕ	6F	6E	6D	6C	6B	69	68	67	65	64	63	61	6ϕ	5E	5D	5B	5A
16914	58	57	55	53	52	5ϕ	4E	4C	4B	49	47	45	43	41	40	3E	3C	3A	38	36	34	32	3ϕ	2E	2B
16939	29	27	25	23	21	1F	1D	1A	18	16	14	12	ϕF	ϕD	ϕB	ϕ9	ϕ7	ϕ4	ϕ2	ϕϕ	FE	FC	F9	F7	F5
16964	F3	F1	EE	EC	EA	E8	E6	E3	E1	DF	DD	DB	D9	D7	D5	D2	Dϕ	CE	CC	CA	C8	C6	C4	C2	C1
16989	BF	BD	BB	B9	B7	B5	B4	B2	Bϕ	AE	AD	AB	A9	A8	A6	A5	A3	A2	Aϕ	9F	9D	9C	9B	99	98
17014	97	95	94	93	92	91	9ϕ	8F	8E	8D	8C	8B	8A	89	89	88	87	87	86	85	85	84	84	83	83
17039	83	82	82	82	81	81	81	81	81	81	81	81	81	81	81	82	82	82	83	83	83	84	84	85	85
17064	86	87	87	88	89	89	8A	8B	8C	8D	8E	8F	9ϕ	91	92	93	94	95	97	98	99	9B	9C	9D	9F
17089	Aϕ	A2	A3	A5	A6	A8	A9	AB	AD	AE	B0	B2	B4	B5	B7	B9	BB	BD	BF	C1	C2	C4	C6	C8	CA
17114	CC	CE	Dϕ	D2	D5	D7	D9	DB	DD	DF	E1	E3	E6	E8	EA	EC	EE	F1	F3	F5	F7	F9	FC	FE	FF
17139	21	ϕ4	ϕϕ	11	E5	ϕ1	CD	8B	45	21	ϕD	ϕϕ	11	AB	ϕ2	CD	AB	ϕ2	CD	A9	45	21	19	ϕϕ	C3
17164	67	44	CD	67	44	21	ϕ4	ϕϕ	11	E5	ϕ1	CD	A9	45	21	ϕD	ϕϕ	11	AB	ϕ2	CD	A9	45	21	19
17189	ϕϕ	11	35	ϕ1	3E	ϕ3	CD	B9	45	21	1E	ϕϕ	11	7F	ϕ1	3E	ϕ3	CD	B9	45	21	2D	ϕϕ	11	46
17214	ϕ1	3E	ϕ4	CD	B9	45	21	28	ϕϕ	11	4A	ϕ2	3E	ϕ3	CD	B9	45	21	2D	ϕϕ	11	46	ϕ2	3E	ϕ3
17239	CD	B9	45	21	32	ϕϕ	11	EC	ϕ1	3E	ϕ5	CD	14	46	21	37	ϕϕ	11	B2	ϕ2	3E	ϕ4	CD	14	46
17264	21	3C	ϕϕ	11	D1	ϕ1	3E	ϕ5	CD	14	46	21	41	ϕϕ	11	97	ϕ2	3E	ϕ4	CD	14	46	ϕ6	ϕ8	CD
17289	66	46	ϕ6	ϕA	CD	89	46	ϕ6	ϕ1	CD	8D	46	ϕ6	ϕA	CD	89	46	ϕ6	ϕ1	CD	8D	46	ϕ6	ϕA	C3
17314	89	46	CD	BB	ϕ2	22	82	4ϕ	3E	ϕϕ	32	84	4ϕ	ϕ1	ϕϕ	ϕϕ	21	A5	ϕϕ	CD	54	45	21	4D	41
17339	5E	23	56	23	EB	CD	92	45	EB	7E	23	5E	23	56	23	2ϕ	ϕ8	E5	26	ϕϕ	6F	CD	9D	45	E1
17364	7E	FE	FF	2ϕ	E2	C3	92	44	21	19	ϕϕ	CD	16	45	CD	6F	44	21	92	4ϕ	CD	4A	46	21	19
17389	ϕϕ	11	55	ϕϕ	CD	Bϕ	44	44	4D	21	19	ϕϕ	C3	54	45	21	2D	ϕϕ	CD	16	45	CD	6F	44	21
17414	EB	4ϕ	CD	4A	46	21	2D	ϕϕ	11	9B	ϕϕ	CD	Bϕ	44	E5	CD	36	44	21	74	ϕϕ	ED	4B	1ϕ	4ϕ
17439	ϕ9	7E	E1	ϕ1	B4	ϕϕ	54	5D	A7	FA	33	44	ED	42	Dϕ	EB	ϕ9	C3	36	44	ED	42	D8	44	4D
17464	21	2D	ϕϕ	C3	54	45	21	2D	ϕϕ	CD	16	45	C5	21	19	ϕϕ	CD	16	45	E1	B7	ED	42	CD	BD
17489	44	CD	6D	44	21	34	41	C3	4A	46	CD	3E	44	CD	FC	43	CD	A4	43	C3	11	43	CD	DC	43
17514	C3	A4	43	44	4D	CD	8ϕ	44	C5	21	5A	2—	CD	87	45	F1	21	5F	ϕϕ	C3	87	45	21	8A	41
17539	09	7E	21	5A	ϕϕ	ϕϕ	CD	BD	44	ϕ1	8A	41	ϕ9	46	C9	ED	4B	86	4ϕ	ϕB	ED	43	86	4ϕ	78
17564	21	41	ϕϕ	A7	F2	87	45	ϕ3	ED	43	86	4ϕ	3E	ϕϕ	21	32	ϕϕ	C3	87	45	D5	CD	16	45	E1
17589	C5	CD	16	45	E1	B7	ED	42	ϕ1	68	ϕ1	CB	7C	28	ϕ2	ϕ9	C9	E5	B7	ED	42	3ϕ	ϕ2	E1	C9
17614	C1	C9	7E	32	82	4ϕ	23	CD	DE	44	2A	82	4ϕ	C3	FC	44	7E	FE	EF	28	ϕB	23	E5	6F	26
17639	ϕϕ	CD	F2	44	E1	18	Fϕ	22	84	4ϕ	E9	CD	ϕϕ	45	ED	Bϕ	ED	53	1C	4ϕ	C9	CD	ϕ9	45	EB
17664	B7	ED	42	22	1C	4ϕ	ED	Bϕ	C9	ED	5B	1ϕ	4ϕ	19	ED	5B	1C	4ϕ	ϕ1	ϕ5	ϕϕ	C9	ED	4B	1ϕ
17689	4ϕ	ϕ9	7E	FE	9ϕ	ϕ1	FF	7F	Dϕ	D6	8ϕ	ϕ1	ϕϕ	ϕϕ	D8	47	23	7E	F5	F6	8ϕ	57	23	5E	21
17714	ϕϕ	ϕϕ	CB	23	CB	12	CB	15	CB	14	10	F6	44	4D	F1	CB	7F	C8	21	ϕϕ	ϕϕ	B7	ED	42	44
17739	4D	C9	D5	CD	16	45	D1	79	C9	ED	5B	1ϕ	4ϕ	19	11	ϕϕ	ϕϕ	78	B1	28	1B	CB	78	28	ϕ7
17764	E5	CD	44	45	E1	16	8ϕ	1E	91	CB	21	CB	1ϕ	1D	3ϕ	F9	CB	38	CB	19	78	B2	47	73	CD
17789	82	45	ϕ1	ϕϕ	ϕϕ	23	7ϕ	23	71	C9	ϕ6	ϕϕ	4F	CB	7F	28	C6	ϕ6	FF	18	C2	ED	4B	82	4ϕ
17814	7C	Aϕ	67	7D	A1	B4	C9	E5	CD	4D	45	E1	BA	C8	7B	81	C3	87	45	CD	ϕB	46	ED	4B	1ϕ
17839	4ϕ	ϕ9	4E	ϕ6	ϕϕ	23	23	ED	Bϕ	C9	4F	ϕ6	ϕϕ	D5	C5	CD	16	45	E1	D1	D5	E5	CD	D3	45
17864	C1	D1	CD	ϕB	46	21	61	4ϕ	ED	B8	C9	6ϕ	69	11	1ϕ	27	CD	ϕϕ	46	32	5D	4ϕ	11	E8	ϕ3
17889	CD	ϕϕ	46	32	5E	4ϕ	11	64	20	CD	7ϕ	46	32	5F	40	11	ϕA	ϕϕ	CD	ϕϕ	46	32	6ϕ	40	7D
17914	C6	1C	32	61	4ϕ	C9	F6	FF	ED	52	3C	3ϕ	FB	19	C6	1C	C9	E5	2A	ϕC	4ϕ	19	54	5D	E1
17939	C9	F5	CD	4D	45	2A	ϕC	4ϕ	19	F1	47	3E	ϕϕ	11	21	ϕϕ	A7	CB	79	2ϕ	1B	CB	39	3ϕ	ϕ1
17964	3C	ϕC	A7	ϕD	28	ϕ7	36	ϕ8	ED	52	1ϕ	F7	C9	3D	2ϕ	ϕ7	37	ϕ9	ED	52	1ϕ	ϕ1	C9	36	ϕϕ
17989	ED	52	1ϕ	FA	C9	AF	32	83	4ϕ	CD	Dϕ	44	2A	84	4ϕ	ϕ1	FF	ϕϕ	3E	34	ED	B1	7E	FE	C9
18014	Cϕ	23	7E	FE	FF	2ϕ	E9	C9	ϕE	ϕϕ	ED	43	84	4ϕ	21	55	ϕϕ	CD	16	45	41	C5	21	5ϕ	ϕϕ
18039	CD	16	45	D1	CB	22	CB	22	79	92	4F	16	ϕϕ	2A	ϕC	40	23	C9	79	CD	9E	46	7A	83	57
18064	CB	2F	CB	2F	81	4F	3E	ϕ3	A2	57	23	1ϕ	EC	C9	D5	C5	E5	ED	4B	84	4ϕ	11	21	ϕϕ	A7
18089	FA	C6	46	CB	2F	3ϕ	ϕ1	ϕC	3D	FA	BC	46	36	88	19	1ϕ	F7	18	ϕF	ϕD	2ϕ	ϕ7	36	8A	19
18114	1ϕ	ϕ2	18	ϕ5	36	8ϕ	19	1ϕ	FB	E1	C1	D1	C9	3C	ϕ1	28	ϕ1	14	ϕϕ	ϕ4	ϕ8	ϕD	12	1B	1F
18139	26	2D	34	3B	3F	48	4D	52	56	5A	14	ϕϕ	ϕ2	ϕ1	ϕϕ	ϕ1	ϕϕ	ϕ1	ϕϕ	ϕ1	ϕϕ	FF	ϕϕ	FF	FF
18164	ϕ1	ϕϕ	ϕ1	ϕϕ	ϕ1	ϕϕ	FD	FF	ϕϕ	FF	FF	ϕϕ	FF	ϕϕ	FF	ϕϕ	FF	ϕϕ	FF	ϕϕ	FF	ϕϕ	FF	FF	FF
18189	ϕϕ	ϕϕ	ϕ1	ϕϕ	ϕ1	FF	ϕϕ	FF	ϕϕ	FF	ϕϕ	ϕ3	ϕ1	ϕϕ	ϕ1	ϕ1	ϕϕ	ϕ1	ϕϕ	ϕ1	ϕϕ	7F	7F	7F	7F
18214	7F	7F	7F	7F	7F	7F	7F	7F	7F	7F	7F	7F	7F	7F	7F	7F	7F	7F	7F	7F	7F	7F	7F	7F	7F
18239	7F	7F	7F	7F	7F	7F	7F	7F	7F	7F	7F	7F	7F	7F	7F	7F	7F	7F	7F	7F	7F	7F	7F	7F	7F
18264	7F	7F	7F	7F	7F	7F	ϕ8	2E	ϕ2	3C	3C	CD	83	47	1ϕ	FB	49	ϕF	F8	ϕ3	FE	FF	6A	ϕ2	ϕϕ
18289	ϕϕ	ϕϕ	ϕϕ	5F	F2	Eϕ	FF	Cϕ	FF	ϕ7	3E	C3	91	47	21	F9	ϕϕ	11	D7	ϕ2	CD	A9	45	21	1F
18314	ϕ1	11	F8	ϕ2	C3	A9	45	CB	ϕ8	43	CD	B8	47	ED	4B	5E	47	CD	45	48	21	55	ϕϕ	CD	16
18339	45	C5	21	55	ϕϕ	ϕ1	ϕϕ	ϕ—	CD	54	45	CD	ϕB	43	C1	21	55	ϕϕ	CD	54	45	CD	E3	47	ϕ6
18364	16	CD	66	46	ϕ6	2ϕ	CD	89	46	21	22	47	ϕ6	1E	CD	7E	48	21	1E	ϕϕ	11	E7	ϕ2	3E	ϕ8
18389	CD	B9	45	21	23	ϕϕ	11	Fϕ	ϕ2	3E	ϕ4	C3	B9	45	3E	ϕ1	32	21	4ϕ	21	73	ϕϕ	CD	27	48
18414	22	88	4ϕ	21	78	ϕϕ	CD	27	48	22	8A	4ϕ	21	23	ϕϕ	CD	16	45	CB	78	28	19	21	AA	ϕϕ
18439	CD	16	45	ϕ3	21	AA	ϕϕ	CD	54	45	ϕ1	ϕϕ	ϕϕ	21	23	ϕϕ	CD	54	45	ϕ1	ϕϕ	ϕϕ	21	ϕA	ϕϕ
18464	ϕ9	22	9ϕ	4ϕ	C3	B4	48	CD	16	45	21	ϕϕ	ϕϕ	B7	ED	42	CB	78	28	ϕ1	44	3E	63	B8	Dϕ
18489	AF	32	21	4ϕ	C9	CD	4E	48	B1	C3	76	48	CD	4E	48	2F	B1	2F	C3	76	48	3E	ϕ1	CB	39
18514	3ϕ	ϕ1	ϕ7	CB	38	3ϕ	ϕ2	ϕ7	ϕ7	26	ϕϕ	68	E5	ϕ6	ϕ5	29	1ϕ	FD	D1	19	ϕ9	4F	ED	5B	ϕC
18539	4ϕ	19	23	7E	CB	7F	28	ϕ2	EE	ϕF	C9	CB	5F	28	ϕ2	EE	8F	77	C9	3A	21	4ϕ	B7	C8	C5
18564	4E	23	46	23	3E	2B	B8	38	ϕA	3E	3F	B9	38	ϕ5	E5	CD	45	48	E1	C1	1ϕ	E9	C9	78	3C
18589	28	ϕB	3D	28	ϕD	17	3E	7F	3ϕ	ϕ2	3C	3C	C9	81	F8	3E	81	C9	81	Fϕ	3E	7F	C9	C1	5ϕ
18614	ϕϕ	CD	16	45	79	3C	32	7A	47	21	19	ϕϕ	CD	19	45	CD	8ϕ	ϕ9	CD	BD	44	44	4D	CD	8ϕ
18639	21	55	ϕϕ	CD	16	45	21	DC	46	ϕ9	4E	ϕ6	ϕϕ	21	3B	ϕ1	ϕ9	CD	BD	44	44	4D	CD	8ϕ	44
18664	32	8E	4ϕ	78	32	8F	4ϕ	ED	5B	88	4ϕ	2A	8A	4ϕ	ED	4B	8C	4ϕ	CD	46	4A	22	74	47	2A
18689	88	4ϕ	ED	5B	8A	4ϕ	ED	4B	8C	4ϕ	78	ED	44	47	CD	46	4A	E5	ED	5B	9ϕ	4ϕ	ED	4B	8E
18714	4ϕ	79	48	ED	44	47	CD	46	4A	22	6E	47	E1	ED	5B	9ϕ	4ϕ	ED	4B	8E	4ϕ	79	ED	44	4F
18739	CD	46	4A	22	68	47	3A	8C	4ϕ	ED	44	6F	3A	8E	4ϕ	CD	55	4A	22	72	47	3A	8D	4ϕ	ED
18764	44	6F	3A	8F	4ϕ	CD	5C	5A	22	6A	47	3A	8C	4ϕ	6F	3A	8F	4ϕ	CD	55	4A	22	6C	47	3A
18789	8D	4ϕ	6F	3A	8E	4ϕ	CD	5C	4A	22	7C	47	3A	8C	4ϕ	6F	CD	82	4A	22	76	47	3A	8D	4ϕ
18814	6F	CD	76	4A	22	78	47	3E	1E	32	82	4ϕ	11	E6	46	21	22	47	E5	1A	13	D5	B7	28	28
18839	4F	2A	68	47	ED	5B	6A	47	CD	63	4A	22	68	47	2A	6E	47	ED	5B	7ϕ	47	CD	63	4A	22
18864	6E	47	2A	74	47	ED	5B	76	47	CD	63	4A	22	74	47	D1	1A	13	D5	B7	28	28	4F	2A	68

18889	47	ED	5B	6C	47	CD	63	4A	22	68	47	2A	6E	47	ED	5B	72	47	CD	63	4A	22	6E	47	2A
18914	74	47	ED	5B	78	47	CD	63	4A	22	74	47	2A	74	47	CB	7C	28	ϕ7	3E	7F	32	83	4ϕ	18
18939	36	ϕ6	ϕ7	AF	B4	28	ϕB	ϕ5	CB	3F	28	ϕ4	CB	1D	18	F7	CB	1D	4D	C5	ED	5B	6E	47	CD
18964	8D	4A	7D	CD	E4	4A	47	3A	7A	47	8ϕ	32	83	4ϕ	C1	ED	5B	68	47	CD	8D	4A	79	CD	E4
18989	4A	ED	44	C6	1F	D1	E1	77	23	3A	83	4ϕ	77	23	3A	82	4ϕ	3D	32	82	4ϕ	C2	9ϕ	49	C9
19014	E5	C5	79	CD	9E	4A	F1	D1	E5	CD	9E	4A	D1	19	C9	CD	76	4A	EB	C3	9E	4A	CD	82	4A
19039	EB	C3	9E	4A	79	A7	FA	6D	4A	47	19	1ϕ	FD	C9	ED	44	47	B7	ED	52	1ϕ	FB	C9	26	ϕϕ
19064	CB	7D	28	ϕ1	25	CB	2C	CB	1D	C9	26	ϕϕ	CB	7D	28	ϕ1	25	29	29	29	C9	78	B7	C8	EB
19089	ED	6A	EA	9A	4A	1ϕ	F9	EB	C9	11	FF	7F	C9	A7	F2	AB	4A	21	ϕϕ	ϕϕ	B7	ED	52	EB	ED
19114	44	4B	5A	16	ϕϕ	47	CB	7B	28	ϕ1	15	62	CB	21	CB	13	CB	12	79	17	7B	6A	17	CB	15
19139	CB	14	81	4F	ED	5A	EB	21	ϕϕ	ϕϕ	78	ϕ6	ϕ7	1F	3ϕ	ϕ8	19	CB	21	3ϕ	ϕ5	23	18	ϕ2	CB
19164	21	CB	13	CB	12	1ϕ	ED	C9	A7	28	34	42	ϕE	ϕϕ	CB	1ϕ	3ϕ	ϕ9	21	ϕϕ	ϕϕ	B7	ED	52	ϕD
19189	18	ϕ1	EB	57	29	7C	8A	3ϕ	1D	1E	ϕϕ	CB	3A	CB	18	ϕ6	ϕ7	B7	ED	52	3ϕ	ϕ6	19	29	1ϕ
19214	F7	18	ϕ4	29	2C	1ϕ	F1	7D	ϕC	Cϕ	ED	44	C9	3E	7F	C9	CD	7C	47	2A	88	4ϕ	ED	5B	8A
19239	4ϕ	CB	7C	28	ϕF	44	4D	21	ϕϕ	ϕϕ	B7	ED	52	EB	21	ϕϕ	ϕϕ	B7	ED	42	ϕ1	Aϕ	ϕF	B7	ED
19264	42	D8	ϕ6	ϕ3	CB	3C	CB	1D	1ϕ	FA	7C	FE	ϕ8	Dϕ	3A	8B	4ϕ	CB	7F	28	ϕ2	ED	44	FE	ϕ4
19289	Dϕ	3A	91	4ϕ	FE	ϕ4	Dϕ	EB	ϕ6	ϕ5	29	1ϕ	FD	3E	ϕ9	D5	CD	9B	4B	D1	32	5E	47	D5	3E
19314	13	CD	E4	4A	26	ϕϕ	6F	ϕ6	ϕ3	29	1ϕ	FD	ED	5B	9ϕ	4ϕ	B7	ED	52	D1	ϕ6	ϕ6	29	1ϕ	FD
19339	3E	2D	CD	9B	4B	32	5F	47	47	3A	5E	47	4F	C3	3E	48	CB	7C	2ϕ	ϕ6	B7	ED	52	D8	3D
19364	C9	19	D8	3C	C9	21	FE	1F	7E	2B	E6	7F	4F	7E	E6	3F	FE	2F	38	ϕ2	D6	2ϕ	8ϕ	47	CB
19389	19	E5	38	ϕ5	CD	3E	48	18	ϕ3	CD	45	48	E1	7C	B7	2ϕ	DE	C9	CD	1C	1C	1C	1C	1C	1C
19414	1C	1C	1C	1C	1C	1C	1C	1C	1C	1C	1C	1C	1C	1C	1C	1C	1C	1C	1C	1C	1C	1C	1C	1C	1C
19439	1C	1C	P																						

DESCRIÇÃO DOS SÍMBOLOS GRÁFICOS UTILIZADOS

Linhas:

5032 . . . GRAPHICS + (21) SHIFT A. . .

5515 . . . GRAPHICS + H + D + G + : . . .

7090 . . . GRAPHICS + SPACE + I + L + S . . . GRAPHICS + SHIFT 8 + (25) SPACE. . .

7091 . . . GRAPHICS + (4) SPACE + SHIFT 6 + SHIFT Q + (7) SPACE + K + M + / + H + (5) SPACE + M + T + S + (7) SPACE . . .

8005 . . . GRAPHICS + (9) SPACE + R + D + F + (20) SPACE. . .

8006 . . . GRAPHICS + (12) SPACE + CABINE + (14) SPACE. . .

8010 . . . (256) SPACE

8020 . . . GRAPHICS + (32) SHIFT 6 + (22) SPACE + M + T + S + (7) SPACE + SHIFT K + SPACE + SHIFT J + (6) SPACE + SHIFT R + (17) SPACE + M + SPACE + SHIFT J + (2) SPACE + H + SPACE + SHIFT J + (18) SPACE + KM/H + (2) SPACE + O (LETRA) + SPACE + SHIFT J + (2) SPACE + O (LETRA) + SPACE + SHIFT L + (2) SPACE + SHIFT W + (6) SPACE + SHIFT W + (14) SPACE + T + SPACE + SHIFT J + (2) SPACE + R + SPACE + SHIFT J + (13) SPACE + (9) SHIFT A + (2) SPACE + O (LETRA) + SPACE + SHIFT J + (2) SPACE + SHIFT J + SPACE + SHIFT J + (6) SPACE + SHIFT 8 + (6) SPACE + SHIFT A + BCN + (5) SHIFT A + SPACE + SHIFT J + (6) SPACE + SHIFT R + (2) SPACE + SHIFT E + SHIFT 7 + SHIFT 2 + SHIFT 7 + (3) SPACE + SHIFT R + SPACE + (9) SHIFT A + (16) SPACE + SHIFT 8 + (6) SPACE + SHIFT A + BRG + SHIFT A + DST + SHIFT A + (7) SPACE + C + SPACE + SHIFT J + (5) SPACE + SHIFT E + SHIFT 2 + (6) SPACE + (9) SHIFT A + (2) SPACE + F + SPACE + SHIFT J + SPACE + O (LETRA) + SPACE + SHIFT J + (13) SPACE + (9) SHIFT A + (2) SPACE + L + SPACE + SHIFT J + SPACE . . .

8030 . . . GRAPHICS + SPACE + M + SPACE + SHIFT J + (2) SPACE + SHIFT W + (6) SPACE + SHIFT W + (14) SPACE + A + SPACE + SHIFT J + (2) SPACE + B + SPACE + SHIFT J + (6) SPACE + SHIFT R + (6) SPACE + TREM + (7) SPACE + P + SPACE + SHIFT J + (33) SPACE. . .

8105. . . GRAPHICS + (24) SPACE + ESCALA + (2) SPACE . . .

8106 . . . GRAPHICS + (5) SPACE + MAPA + SPACE + DE + SPACE + NAVEGA½AO + (2) SPACE . . .

8110 . . . GRAPHICS + (15) SPACE + N + (30) SPACE + 000 + (68) SPACE + SHIFT R + (15) SPACE + SHIFT R + ONW + (12) SPACE + SHIFT 8 + (31) SPACE + SHIFT 8 + (4) SPACE + SHIFT R + ONE + (23) SPACE + SHIFT T + (31) SPACE + SHIFT 5 + (10) SPACE . . .

8120 . . . GRAPHICS + (21) SPACE + SHIFT 5 + SHIFT R + (10) SPACE + W + (10) SPACE + SHIFT R + SPACE + (2) SHIFT G + (2) SPACE + SHIFT R + (3) SPACE + SHIFT 8 + (7) SPACE + E + SPACE + 270 + (8) SPACE + IW + (4) SPACE + IE + (3) SPACE + SHIFT 8 + (6) SPACE + 090 + (22) SPACE + SHIFT 8 + (30) SPACE + SHIFT E + SHIFT 8 + (30) SPACE + SHIFT 5 + (32) SPACE + SHIFT 8 + (30) SPACE + SHIFT Y + (10) SPACE . . .

8130 . . . "GRAPHICS + (17) SPACE + SHIFT R + OSE + (13) SPACE + SHIFT R + OSW + (82) SPACE + 0 + (4) SPACE + 5 + (16) SPACE + 180 + (7) SPACE + SHIFT R + (17) SPACE + S + (8) SPACE + (5) SHIFT 6 + SHIFT Q + (2) SPACE (2) SPACE + GRAPHICS"
(esta linha apresenta erro na listagem)

8220 . . . GRAPHICS + (7) SPACE + SIMULADOR + SPACE + DE + SPACE + VOO + (9) SPACE . . .

8230 . . . GRAPHICS + SHIFT 8 . . .

8232 . . . GRAPHICS + (32) SHIFT 7 . . .

8234 . . . GRAPHICS + SHIFT 8

8236 . . . GRAPHICS + SHIFT 8

8238 . . . GRAPHICS + (15) SHIFT 6 + SHIFT 8 . . .
. . .GRAPHICS + SHIFT 8 . . . .

8240 . . . GRAPHICS SHIFT 8 . . .

Obs.: As linhas 8110, 8120 e 8130 possuem antes das aspas (20) SPACE.

```
1500 GOTO 8000
2010 LET T=USR 17166
2100 IF C(3)<0 THEN GOTO 3000
2105 IF C(19)+X(BN)>7.5 AND C(3)
<1500 THEN GOTO 3020
2110 IF C(2)+2*C(7)<90 THEN GOTO
 3100
2115 IF C(7)*C(2)>1200 THEN GOTO
 3040
2340 PLOT C(10),C(11)
2345 LET T=USR 17160
2346 UNPLOT C(10),C(11)
2410 IF C(29)=VAL "0" THEN GOTO
2000
2430 IF C(29)=VAL "2" THEN GOSUB
 5800
2440 IF C(29)=VAL "4" THEN GOTO
3100
2450 IF C(29)=VAL "8" THEN GOTO
7000
2460 IF C(29)=VAL "1" THEN GOTO
5900
2490 GOTO 2000
3000 LET X$="SOLO"
3010 GOTO 5000
3020 LET X$="MORRO"
3030 GOTO 5000
3040 LET X$="SOLO"
3042 LET Y$="VOCE CAIU PORQUE"
3044 LET Z$="VOCE PERDEU SEUS FL
APS"
3049 GOTO 3105
3100 LET Y$="VOCE PERDEU SUSTENT
ACAO"
3105 LET D=1
3110 LET T=USR 17169
3120 LET C(12)=4*RND
3130 LET C(13)=C(13)+D
3132 IF ABS C(13)>8 THEN LET D=-
D
3140 LET C(3)=C(3)-77
3180 IF C(3)>0 THEN GOTO 3110
3190 IF C(19)<7.5 THEN GOTO 3000
3195 GOTO 3020
3200 LET Y$="VOCE PERDEU SUSTENT
ACAO"
3205 LET D=VAL "1"
3210 LET T=USR 18360
3220 LET C(12)=4*RND
3230 LET C(13)=C(13)+D
3232 IF ABS C(13)>8 THEN LET D=-
D
3240 LET C(3)=C(3)-29
3280 IF C(3)>VAL "0" THEN GOTO 3
210
3290 LET X$="AEROPORTO"
3295 GOTO 5000
4000 GOSUB 8200
4010 PRINT "VOCE PREFERE APENAS"
4014 PRINT "A FASE DE ATERRISSAG
EM?"
4020 GOSUB 4900
4030 PRINT
4040 IF INKEY$="S" THEN GOTO 405
0
4042 IF INKEY$="N" THEN GOTO 410
0
4044 GOTO 4040
4050 PRINT "OK,SOMENTE A ATERRIS
4060 LET R=VAL "2"
4061 LET A=1.5*PI
4065 LET BN=VAL "1"
4066 GOSUB 5825
4070 LET C(3)=800
4075 LET C(1)=VAL "90"
4080 LET T=2200
4090 RETURN
4100 PRINT "OK,VOO COMPLETO"
4110 LET R=10+RND*5
4112 LET A=RND*2*PI
4120 LET BN=INT (1+RND*6)
4122 GOSUB 5800
4130 LET C(1)=INT (RND*360)
4150 LET T=2000+INT (RND*999)
```

INSTRUÇÕES PARA JOGAR O SIMULADOR

Assim que você tenha pressionado *NEW LINE*, imediatamente surgirá no vídeo a primeira pergunta: "Você prefere apenas a fase de aterrisagem?"

Se você responder *"S"* o programa somente lhe mostrará a fase final de aproximação do aeroporto, o pouso visual ou aterrisagem. Se você responder *"N"* o computador rodará o programa completo, com todos os problemas de navegação.

A segunda pergunta será: "Você deseja incluir os efeitos do vento (S/N)?" Isto significa que se você quiser pilotar com os efeitos do vento sobre o avião deve digitar "S". Porém, só aconselhamos jogar com os efeitos do vento os pilotos experimentados.

Existem três diferentes apresentações na tela do programa:

Cabine de Comando

A tela mostra a cabine de comando com todos os instrumentos essenciais para pilotar um avião, na metade inferior da tela, e uma visão exterior através da janela frontal, na metade superior da tela. Depois de responder às perguntas do início do programa, automaticamente, aparecerá no vídeo a cabine de comando e seus instrumentos. A metade superior da tela mostra a janela frontal da cabine, através da qual se vê o horizonte formado pela junção de uma parte clara (céu) e uma parte escura (terra). Quando o ângulo do avião em relação ao solo mudar, mudará também a visão do horizonte mostrando os efeitos de "mergulho" e "subida".

Na parte inferior da tela, abaixo da janela, está situado o painel de instrumentos com os seguintes mostradores:

- Motor — Controlado pelo acelerador (manche)
- Flap — Um mostrador indica os ângulos dos flaps nas asas.
- Combustível (COMB) — Neste mostrador a área sombreada mostra a quantidade de combustível que você dispõe a cada momento.
- Horizonte (+ HOR —) — Medidor de ângulo de subida ou descida, mostra a velocidade de subida (positiva) ou de descida (negativa) do avião, graduado em unidades de 600 m por minuto. Quando o avião estiver nivelado, o indicador HOR deverá estar precisamente na metade do mostrador.
- Altímetro (MTS) — Os números indicam a altura do avião acima do nível do mar, medida em MTS.
- Velocidade (KM/H) — Os números indicam a velocidade do avião no ar em KM/H.
- RDF (Radio Direction Finder) — É mostrado na metade esquerda do painel de instrumentos. Representa a posição do Rádio Farol que está sendo usado no momento como referência, em relação a um pequeno avião desenhado no meio de um círculo dividido em 360°. A direção em que o avião se desloca é para onde o nariz do pequeno avião aponta. A posição do Rádio Farol em relação ao curso do avião é mostrada por um ponto luminoso intermitente situado em algum lugar no círculo de 360°. Para dirigir o avião para este Rádio Farol basta deslocar o ponto luminoso para o nariz do avião desenhado.

- Curso — Aparecerá em um mostrador logo acima do círculo de RDF; mostra a direção do avião na Rosa dos Ventos, de 0° a 360°
- BCN — Refere-se ao prefixo do Rádio Farol que está sendo usado como referência (vide "Mapa de Navegação").
- BRG — Ângulo do Rádio Farol usado em relação ao curso do avião.
- DST — Distância em milhas náuticas entre o avião e o Rádio Farol usado no momento.

Mapa de Navegação

A tela mostra um mapa de navegação com as posições ocupadas pelo avião: a posição do aeroporto, a posição dos auxílios rádio (Rádio Farol) e as cercanias do aeroporto. Esta tela será obtida pressionando-se a tecla *"M"*, a qual irá apresentar um mapa geral de navegação que contém o aeroporto (um pequeno retângulo acinzentado), uma rosa dos ventos para sua orientação — com os quatro pontos principais: NORTE (N — 0°), LESTE (E — 90°), SUL (S — 180°) e OESTE (W — 270°) e as cercanias do aeroporto em uma área de 32 milhas por 20 milhas. Uma escala no canto inferior direito do vídeo permite fazer medidas. Como o aeroporto fica na direção 90° — 270°, qualquer aterrisagem deverá ser feita neste alinhamento. O mapa mostra também a posição de vários Rádios Faróis nas cercanias da pista de pouso, e seus nomes ou prefixos. Existem 4 Rádios Faróis periféricos (ONE, OSE, OSW e ONW) e 2 Rádios Faróis interiores (IE e IW), que estão mais ou menos a 3 milhas de cada extremidade da pista para auxílio na aproximação final e aterrisagem. Você verá sempre o Rádio Farol escolhido como um ponto luminoso intermitente, porém imóvel. O avião será representado também por um ponto luminoso intermitente, porém sempre se deslocando na tela. O aeroporto está situado perto de uma cadeia de montanhas com a altura de 1500 metros, situada a leste do aeroporto à distância de 7,5 milhas. Cuidado para não passar por esta cadeia a menos de 1550 metros pois você poderá se chocar contra ela. Para voltar à cabine de comando pressione a tecla *"M"* outra vez.

```
4160 LET C(3)=3000+INT (RND*2000
)
4490 RETURN
4510 CLS
4511 LET P$(4)=P$(1)
4512 LET P$(18)=P$(3)
4513 LET P$(39)=P$(2)
4514 LET P$(10)=P$(7)
4515 LET P$(37)=P$(8)
4516 LET P$(109)=P$(21)
4517 LET P$(13)=P$(11)
4518 LET P$(51)=P$(12)
4519 LET P$(41)=P$(34)
4520 PRINT "VOCE DESEJA INCLUIR"
4530 PRINT "OS EFEITOS DO VENTO?
"
4531 LET P$(84)="R"
4532 LET P$(42)="D"
4533 LET P$(40)="T"
4534 LET P$(17)=P$(7)
4535 GOSUB 4900
4540 IF INKEY$="S" THEN GOTO 460
0
4542 IF INKEY$="N" THEN GOTO 455
0
4544 GOTO 4540
4550 PRINT AT 16,0;"OK,SEM OS EF
EITOS DO VENTO"
4560 RETURN
4600 RAND
4602 LET VW=INT (RND*RND*18)+2
4610 PRINT AT 14,0;"VELOCIDADE D
O VENTO: ";VW;" KM/H"
4620 LET AW=INT (RND*360)
4624 PRINT
4625 PRINT "DIRECAO: ";AW;" GRAU
S"
4630 LET AW=AW*PI/180
4632 LET VW=-127*VW
4635 PRINT AT 21,0;"DIGITE QUALQ
.TECLA MENOS BREAK"
4640 IF INKEY$="" THEN GOTO 4640
4690 LET C(17)=VW*SIN AW
4692 LET C(18)=VW*COS AW
```


CONCURSO MICROCOMPUTADOR PESSOAL TK-82C
COMO PARTICIPAR DO CONCURSO:

VISITE-NOS NA III FEIRA INTERNACIONAL DE INFORMÁTICA E CONHEÇA O "MICROMESTRE."

- Escreva uma monografia sobre o tema: "A Atual Política de Informática Brasileira"
- Datilografe em espaço dois, três laudas no máximo
- Envie os trabalhos até o dia 30/09 com os seguintes dados: NOME, ENDEREÇO, PROFISSÃO.
- A Promoção é dividida em duas classes: 10 a 15 anos e a partir de 15 anos.
- Os trabalhos vencedores serão publicados na revista INTERFACE Nº 12 e seus autores ganharão um microcomputador pessoal TK-82C da MICRODIGITAL.
- Anexe à monografia uma autorização para cessão dos direitos de publicação do título
- Mande seu trabalho para: "INTERFACE-CONCURSO MICROCOMPUTADOR PESSOAL" — Estrada do Tindiba, 2380 — Rio de Janeiro - RJ — CEP: 22700.

O PRÓXIMO PODERÁ SER VOCÊ

Relação dos Ganhadores

O Computador e a Educação — Frederico Celestino Pontes — RJ
O Computador e a Medicina — Dilma Trivelli P. Sandrin — BH
A Automação da Sociedade Brasileira — Paulo Henrique de Faria Pimenta — RJ
A Automação Bancária — Juarez Fernandes Ramos — ES

```
4696 RETURN
4900 PRINT
4903 LET P$(54)=P$(3)
4904 LET P$(104)="T"
4905 LET P$(99)="D"
4906 LET P$(102)=P$(99)
4910 PRINT "DIGITE S PARA SIM"
4930 PRINT "          N PARA NAO"
4931 LET P$(43)=P$(36)
4932 LET P$(73)="I"
4933 LET P$(75)="I"
4934 LET P$(100)=P$(36)
4935 LET P$(98)="U"
4936 LET P$(87)="U"
4937 LET P$(24)=P$(11)
4938 LET P$(79)=P$(77)
4939 LET P$(89)="A"
4940 LET P$(16)=P$(1)
4941 LET P$(47)="E"
4942 LET P$(82)="E"
4943 LET P$(49)="S"
4944 LET P$(44)=P$(8)
4945 LET P$(105)=P$(24)
4946 LET P$(103)="A"
4947 LET P$(97)=P$(11)
4948 LET P$(55)="D"
4949 LET P$(46)=P$(35)
4950 LET P$(76)="D"
4952 LET P$(19)=P$(1)
4954 LET P$(22)=P$(4)
4956 LET P$(56)=P$(13)
4958 LET P$(33)=P$(11)
4960 LET P$(50)="O"
4962 LET P$(101)=P$(37)
4964 LET P$(85)="O"
4966 LET P$(90)="O"
4968 LET P$(86)=P$(76)
4970 LET P$(81)="R"
4990 RETURN
5010 GOSUB 6300
5015 LET IS=VAL "5"
5016 GOSUB 5490
5020 PRINT AT 21,0;
5025 PRINT " RELATORIO DE ACIDEN
TE"
5030 SCROLL
5032 PRINT "
"
5034 SCROLL
5036 SCROLL
5040 PRINT "VOCE BATEU CONTRA O
5042 SCROLL
5050 PRINT "COM A VELOCIDADE DE
";INT C(2);" KM/H"
5052 SCROLL
5054 SCROLL
5060 PRINT Y$
5062 SCROLL
5064 SCROLL
5070 PRINT Z$
5080 SCROLL
5090 LET IS=VAL "8"
5100 IF PEEK 16519<99 THEN GOTO
5150
5110 PRINT "VOCE FICOU SEM COMBU
STIVEL"
5120 LET IS=IS-VAL "1"
5410 GOSUB 5490
5470 PRINT "DIGITE RUN PARA VOAR
 NOVAMENTE"
5480 STOP
5490 FOR I=VAL "1" TO IS
5492 SCROLL
5494 NEXT I
5496 RETURN
5515 PRINT AT 1,1;"[illegible]";C(1);
5520 LET XB=2*X(BN)+29.5
5525 LET YB=2*Y(BN)+24
5530 LET XP=INT (2*C(19)+XB+.5)
5535 LET YP=INT (2*C(20)+YB+.5)
5540 IF XP<VAL "0" THEN LET XP=V
AL "0"
5542 IF XP>VAL "63" THEN LET XP=
VAL "63"
5544 IF YP<VAL "0" THEN LET YP=V
```

Aproximação Visual (Pista de Pouso)

A tela mostra a posição do avião relativa ao solo, vista através da cabine de comando, dando uma visão animada das luzes do aeroporto, as quais se aproximam de você à medida que o avião se aproxima. Apresenta ainda alguns instrumentos essenciais para o controle de pouso, se você optou pelo jogo completo e teve êxito em navegar o avião até que tivesse o alinhamento certo com os Rádios Faróis das cabeceiras da pista, e sua altitude, velocidade e ângulo de descida estão corretos. Pressione a tecla *"V"* cerca de 3 milhas antes, para que apareça na tela em perspectiva as luzes da pista e seu aspecto geral. O horizonte pode ainda ser visto à longa distância no topo da tela. Se você precisar baixar mais para o pouso, ajuste a inclinação do nariz, os flaps e a potência do motor para uma aterrisagem suave. Tenha certeza de que o seu trem de aterrisagem está baixado (pressione a tecla *"G"* para baixar ou recolher o trem de aterrisagem).

POUSO POR INSTRUMENTOS

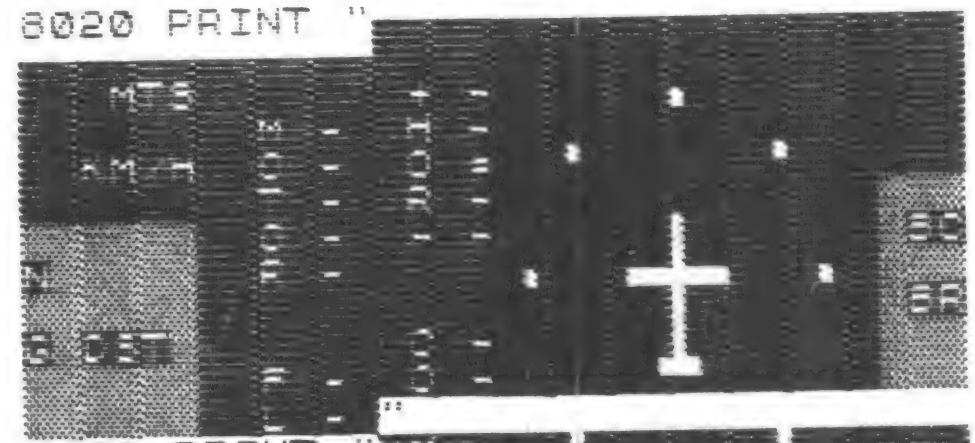
Os instrumentos essenciais de um típico pouso por instrumentos ILS (Instrument Landing System) estão mostrados na parte inferior da tela, estando incluídos: velocidade do avião e altura em que as rodas do avião estão em relação ao solo. No canto inferior esquerdo da tela está um importante instrumento de um ILS: um quadrado indicador que mostra a posição do começo da pista em relação ao avião. Para um piloto ele se comporta como um horizonte artificial, mas em duas dimensões. Quando o avião está precisamente no curso certo (posição e altura) um quadrado preto menor, intermitente, estará no centro do quadrado branco maior. Se o quadrado preto estiver acima do centro, o avião estará descendo em um ângulo muito forçado. Se o quadrado preto estiver abaixo do centro, o avião não estará descendo com um ângulo suficiente. Se o quadrado preto está à esquerda do branco, isto indica que a pista está para a esquerda do rumo do avião, e você deverá corrigir para o lado esquerdo. A mesma coisa deve ser feita quando for para o lado direito.

CONTROLES PARA A PILOTAGEM

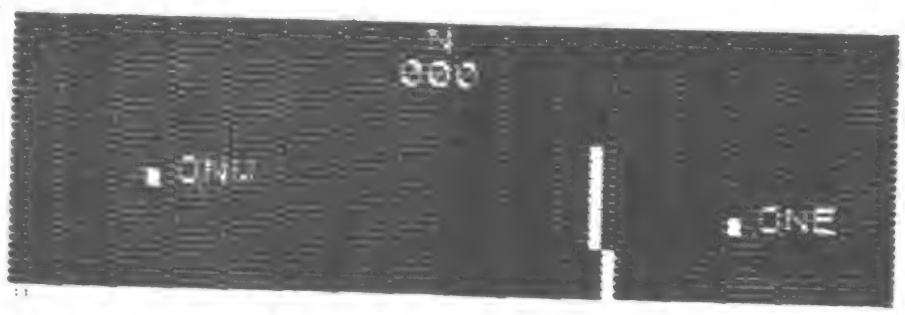
- Manche — O manche do avião é representado pelas teclas 5, 6, 7 e 8.
  Pressione (tecla 5) para inclinar para a esquerda
  Pressione (tecla 8) para inclinar para a direita
  Pressione (tecla 6) para inclinar o nariz do avião para cima.
  Pressione (tecla 7) para inclinar o nariz do avião para baixo.
- Motor — A potência do motor é controlada pelas teclas *"P"* e *"O"*.
  Pressione (tecla *"P"*) para aumentar a potência do motor
  Pressione (tecla "O")para diminuir a potência do motor.
- Flaps — Os flaps das asas são controlados pelas teclas *"F"* e *"D"*.
  Pressione (tecla *"F"*) para baixar os flaps (pressione várias vezes para obter o ângulo desejado).
  Pressione (tecla *"D"*) para retrair os flaps (tantas vezes quanto forem necessárias). Os flaps podem ser baixados ou retraídos em um variado grau, como é mostrado no seu indicador, e só devem ser totalmente baixados para o estágio final de pouso para evitar que o avião perca sustentação e estole. Por outro lado, se baixar os flaps com o avião em alta velocidade, poderá fazer com que sua resistência ao ar cause sérias avarias às asas.

```
AL "0"
5548 IF YP>43 THEN LET YP=43
5610 PLOT XB,YB
5616 UNPLOT XP,YP
5620 LET T=USR 17163
5630 UNPLOT XB,YB
5635 PLOT XP,YP
5640 IF C(3)<VAL "0" THEN GOTO 3
000
5641 IF C(19)+X(BN)>7.5 AND C(3)
5642 IF C(2)+2*C(7)<90 THEN GOTO
 3100
5650 IF C(29)<>VAL "2" THEN GOTO
 5700
5660 GOSUB 5800
5670 GOTO 5520
5780 IF C(29)=0 THEN GOTO 5530
5790 GOTO 8000
5810 LET C(19)=C(19)+X(BN)
5815 LET C(20)=C(20)+Y(BN)
5820 LET BN=BN+1
5825 IF BN>VAL "6" THEN LET BN=V
AL "1"
5830 LET B$=N$(BN)
5835 LET C(19)=C(19)-X(BN)
5840 LET C(20)=C(20)-Y(BN)
5850 LET C(4)=VAL "1"
5860 LET C(5)=VAL "45"
5890 RETURN
5900 LET C(21)=1-C(21)
5910 IF C(21)=0 THEN GOTO 5980
5920 LET G$="BAIX"
5930 IF C(2)<170 THEN GOTO 2000
5940 LET Y$="VOCE CAIU PORQUE"
5942 LET Z$=" VOCE PERDEU O TREM
 DE ATERRISS."
5950 GOTO 3105
5980 LET G$="REC."
5990 GOTO 2000
6050 CLS
6060 IF C(30)>VAL "2" THEN GOTO
6100
6062 PRINT
6064 PRINT
6070 PRINT "VOCE ATERRISSOU SUAV
EMENTE"
6080 GOTO 6200
6100 IF C(30)>VAL "5" THEN GOTO
6150
6110 PRINT "VOCE ATERRISSOU AOS
SOLAVANCOS"
6120 GOTO 6200
6150 PRINT "VOCE QUASE SE ESPATI
FOU"
6200 PRINT
6205 IF ABS C(19)>4096 OR ABS C(
20)>64 THEN GOTO 6300
6210 IF ABS C(20)>20 THEN PRINT
"UM POUCO FORA DO CENTRO DA PIST
A"
6215 PRINT
6220 IF ABS C(20)>50 THEN PRINT
"QUASE NA BEIRA DA PISTA"
6225 PRINT
6230 IF ABS C(19)>3700 THEN PRIN
T "VOCE QUASE PASSOU DA PISTA"
6290 GOTO 6800
6300 PRINT "VOCE TAXIOU PARA FOR
A DA PISTA"
6810 PRINT
6820 PRINT
6910 GOTO 5450
7020 LET C(25)=3830*C(25)
7022 LET C(16)=.63*C(16)
7024 LET C(26)=.63*C(26)
7030 LET C(31)=VAL "6"
7032 LET C(12)=C(12)-VAL "6"
7070 LET C(19)=(C(19)+X(BN))*606
0
7072 LET C(20)=(C(20)+Y(BN))*606
0
7090 LET L$="[illegible]"
7091 LET M$="[illegible]"
```

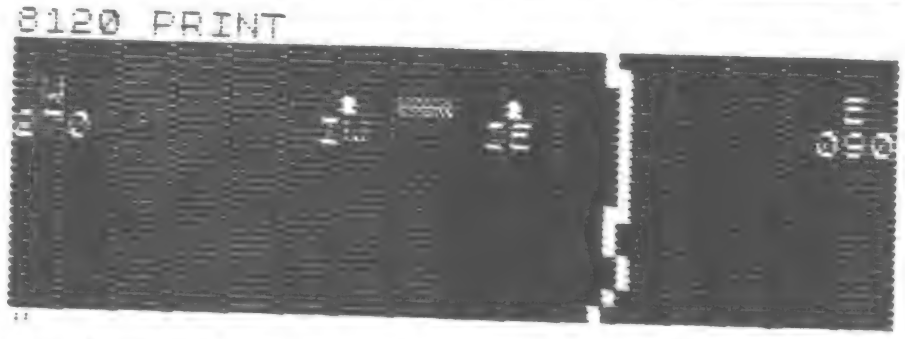
```
7095 LET T=USR 18303
7110 LET T=USR 19230
7130 IF C(3)=VAL "0" THEN GOTO 7
900
7150 IF C(2)+2*C(7)<90 THEN GOTO
 7700
7200 IF C(29)=VAL "0" THEN GOTO
7100
7520 LET C(25)=C(25)/3830
7522 LET C(16)=C(16)/.63
7524 LET C(26)=C(26)/.63
7530 LET C(31)=VAL "12"
7532 LET C(12)=C(12)+VAL "6"
7570 LET C(19)=C(19)/6080-X(BN)
7572 LET C(20)=C(20)/6080-Y(BN)
7590 GOTO 8000
7700 IF C(3)<50 THEN GOTO 7710
7702 LET X$="AEROPORTO"
7709 GOTO 3200
7710 IF C(3)<12 THEN GOTO 7740
7720 LET C(12)=1
7730 GOTO 7100
7740 LET C(12)=VAL "5"
7745 GOTO 7100
7800 IF C(21)=1 THEN GOTO 7810
7802 LET X$="AEROPORTO"
7804 LET Y$="VOCE ATERRISSOU DE
BARRIGA"
7809 GOTO 5000
7810 IF ABS C(19)<4096 OR ABS C(
20)<64 THEN GOTO 7820
7812 LET X$="AEROPORTO"
7819 GOTO 5000
7820 IF C(12)>=VAL "5" THEN GOTO
 7830
7822 LET C(12)=VAL "11"-C(12)
7829 GOTO 7100
7830 LET C(32)=VAL "99"
7832 LET C(16)=VAL "0"
7834 LET C(17)=VAL "0"
7836 LET C(18)=VAL "0"
7839 GOTO 7100
7900 IF C(32)=VAL "16" THEN GOTO
 7800
7910 IF C(2)<VAL "3" THEN GOTO 6
000
7920 GOTO 7100
8000 PRINT AT 0,0;
8005 LET L$="[illegible]RDF[illegible]"
8006 LET M$="[illegible]CABINE[illegible]"
8010 PRINT "
8020 PRINT "
```




```
8030 PRINT "[illegible]TRE
[illegible]"
8040 LET T=USR 18303
8090 GOTO 2000
8100 PRINT AT 0,0;
8105 LET L$="[illegible]"
8106 LET M$="[illegible]MAPA DE NAVEG
8110 PRINT
```

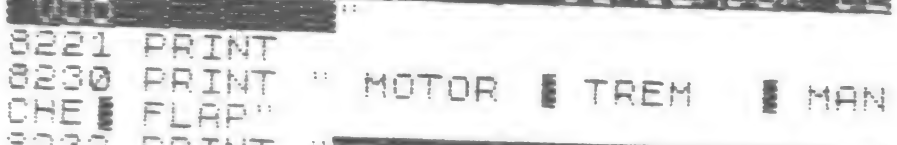


```
"
8120 PRINT
```



```
"
8130 PRINT
```

```
8160 LET T=USR 18303
8190 GOTO 5500
8210 CLS
8220 PRINT "
```



```
"
8221 PRINT
8230 PRINT "  MOTOR ▮ TREM   ▮ MAN
CHE▮ FLAP"
8232 PRINT "
```



```
"
8234 PRINT "O)REDUZ ▮G)BAIXA ▮6)PU
XA  ▮F)AUM."
8236 PRINT "P)AUM.  ▮G)RECOL ▮7)EM
PUR▮D)DIMIN."
8236 PRINT "
```



```
▮5)ES
0.  ▮"
8240 PRINT "M)MAPA-V)VISUAL ▮8)DI
R.  ▮"
8242 PRINT
8290 RETURN
8310 LET T=USR 19369
8320 LET L$=""
8322 LET M$=""
8326 LET T=USR 18303
8390 RETURN
9010 DIM N$(6,3)
9012 DIM X(6)
9014 DIM Y(6)
9020 LET N$(1)="IW "
9021 LET P$(3)="T"
9022 LET X(1)=-2.75
9023 LET P$(35)="D"
9024 LET Y(1)=VAL "0"
9025 LET P$(72)="O"
9026 LET P$(88)="C"
9027 LET P$(77)="A"
9028 LET P$(113)="8"
9029 LET P$(53)="L"
9030 LET N$(2)="IE  "
9031 LET P$(36)="I"
9032 LET X(2)=3.25
9033 LET P$(78)=" "
9034 LET Y(2)=VAL "0"
9040 LET N$(3)="ONW"
9041 LET P$(8)="O"
9042 LET X(3)=-9.75
9043 LET P$(80)=P$(78)
9044 LET Y(3)=VAL "5"
9050 LET N$(4)="ONE"
9052 LET X(4)=11.25
9054 LET Y(4)=VAL "4"
9060 LET N$(5)="OSE"
9061 LET P$(2)="S"
9062 LET X(5)=2.25
9064 LET Y(5)=-7
9069 LET P$(71)="R"
9070 LET N$(6)="OSW"
9071 LET P$(7)="R"
9072 LET X(6)=-12.75
9074 LET Y(6)=-8
9090 RETURN
9100 SAVE "SIMULADO"
9101 RUN
```

Trem de aterrisagem — O trem de aterrisagem é controlado pela tecla *"G"*.
Pressione (tecla *"G"*) para baixar o trem.
Pressione (tecla *"G"*) novamente e o trem será recolhido.
O trem de aterrisagem não deve ser baixado em alta velocidade; se isto for feito você causará sérias avarias no aparelho.

Rádio Farol — Para mudar o Rádio Farol que está sendo usado no momento, basta pressionar a tecla *"B"* e você poderá, seqüencialmente, ir passando para outros Rádios Faróis situados no mapa. A sigla ou prefixo deles aparecerá no mostrador BCN.

Navegação — A parte mais difícil do vôo é a aproximação e pouso. Você pode experimentar deslocar seu avião pelo mapa usando os controles em várias velocidades, altitudes e direções, sem se preocupar imediatamente com os problemas de navegação. Se você deseja, entretanto, pousar o avião terá que aproximá-lo da pista em um curso certo e reto, e tentar pousar num ângulo suave e preciso, sem que tenha que forçar os motores ou a estrutura do avião. Esta é uma tarefa difícil e será necessária muita prática e experiência antes que você possa conseguir aterrisar com todo o sucesso. O mapa e os instrumentos, assim como os rádios-faróis, ajudarão você a determinar sua posição exata e executar as manobras de aproximação do aeroporto para que possa então tentar o pouso. A pista fica em um alinhamento LESTE-OESTE — de 90° a 270° ou vice-versa. Você pode aproximar-se da pista por qualquer de suas extremidades. Como por exemplo: se a sua aproximação for de LESTE para OESTE, você deve levar o avião para uma distância bem ao LESTE da pista. No caso de usar o Rádio Farol IE você precisará manobrar o avião até o Rádio Farol IE estar em uma linha de 270°. Quando você coincidir seu ponto (avião) com o ponto de IE poderá então fazer uma volta e tomar um curso reto sobre uma linha imaginária, em direção ao aeroporto, e quando chegar mais perto poderá passar para *"V"* (visual) e tentar pousar. Em qualquer ocasião você poderá usar qualquer dos outros Rádios Faróis para um objetivo de vôo escolhido. Basta pressionar a tecla "B" para a mudança em seqüência dos prefixos. Sempre que possível mantenha os dígitos de curso iguais aos de BRG. Nunca pressione a tecla BREAK.

TECLAS	COMANDOS
↑	Mergulhar
↓	Subir
→	Virar para a direita
←	Virar para a esquerda
P	Aumentar potência do motor
O	Diminuir potência do motor
F	Aumentar ângulo de flap
D	Diminuir ângulo de flap
G	Abaixar e subir o trem de aterrisagem
B	Mudar os Rádios Faróis seqüencialmente
M	Mudar a tela de Cabine de Comando para Mapa de Navegação e vice-versa
V	Mudar a tela para Visual da Pista de Pouso (*M* para voltar a tela da Cabine de Comando)

UMA REVISTA TÉCNICA É UM VEÍCULO DE FORMAÇÃO E ATUALIZAÇÃO PROFISSIONAL

POR QUE ASSINAR INTERFACE?

Você consolida sua formação técnica atualizando-se sobre: hardware, software, teleprocessamento, mercado de computadores, periféricos, controladores/interface, novos produtos, dispositivos LSI, software básico, assembler, CP/M e novas linguagens de programação.
Você participa de uma revista brasileira eminentemente técnica, formativa, sem entrevistas, didática, neutra e escrita por especialistas da área; e ainda tem descontos especiais em cursos, seminários e congressos promovidos pela revista INTERFACE.

Envie-nos xerox do cupom acompanhado de cheque nominal ou vale postal.

CADASTRO PARA ASSINANTE

Escolha a sua opção de assinatura e marque com um X

☐ 1 ano – Cr$ 7.500,00
☐ 2 anos – Cr$ 15.000,00

Estou enviando cheque nominal à Prodigt – Processamento, Tecnologia e Comunicação Ltda.
Estrada do Tindiba, 2380 – CEP 22700 – Rio de Janeiro – RJ

NOME: _____

ENDEREÇO: _____

_____

EMPRESA: _____

CARGO: _____

CIDADE: _____ ESTADO: _____

CEP: _____ TEL: _____

O endereço acima é: ☐ do trabalho
☐ da casa

MICRO COMPUTADORES

ARQUIVO DE DADOS EM DISCO
VERSUS
ARQUIVO DE DADOS EM CASSETE

4ª Parte

Em termos comparativos o gravador cassete é similar ao drive de disco, enquanto a fita cassete é similar ao disquete.

Um disco flexível (**floppy**) ou disquete é construído de material similar às fitas cassetes, porém tem uma camada magnética em ambos os lados e tem o formato circular de um disco. Em lugar de ser protegida por uma carcaça plástica rígica como as fitas cassetes, o disquete é protegido por um envelope plástico flexível conhecido como "jaqueta" ou "invólucro", o qual tem uma abertura (fenda) semelhante a existente na carcaça das fitas cassetes, através da qual a cabeça de leitura/escrita do drive (PLAY/ RECORD do gravador) entra em contato com a superfície magnética.

Apresenta também um orifício através do qual o drive pode encaixar e mover o meio magnético, e um corte de proteção que permite ou não a gravação no disco.

A forma circular de um disco permite grandes velocidades de operação. Numa fita cassete, passar do primeiro programa ou bloco de dados para o último significa o enrolamento de centenas e centenas de centímetros de fita, enquanto que num disquete passar de um "fim de programa" para outro significa mover a cabeça por uma ou duas polegadas através do disco.

Os discos flexíveis podem operar em altas velocidades, pois foram projetados para gravação direta de impulsos digitais. Porém, nas fitas cassetes de audio o fluxo de dados deve ser lento para permitir a codificação em "tons" que serão gravados precisamente. Por exemplo, num sistema de microcomputador com disco flexível um programa de 16KB é encontrado e carregado na memória RAM em 4 a 8 segundos. O mesmo programa em fita cassete levaria de 4 a 8 minutos para carregar, não computando o tempo necessário para encontrá-lo desde o início da fita.

Em termos práticos, o que representa uma capacidade de 100.000 bytes?

Se lembrarmos que um byte de capacidade equivale a aproximadamente um caractere, uma capacidade de 100.000 bytes equivale a uma capacidade de armazenamento de 100.000 caracteres.

Uma fita cassete C-10 otimizada armazena 37.740 bytes aproximadamente (vide revista nº 9, 3ª parte deste artigo) e um disco flexível tem uma capacidade de armazenamento entre 80.000 a 500.000 caracteres com um tempo de acesso de 1.500 a 15.000 caracteres por segundo, o que é muito superior a qualquer fita cassete de audio. Por exemplo, se uma página de texto contém 54 linhas por 64 colunas, num total aproximado de 3.500 caracteres (byte), um **minifloppy** de densidade simples armazena 23 páginas e um **floppy** de 8 polegadas de densidade simples armazena aproximadamente 100 páginas. Quanto maior a capacidade de armazenamento maior é o custo do **drive de disco,** embora o custo por byte de armazenamento seja geralmente menor.

A quantidade de espaço de armazenamento é função da aplicação específica e das operações necessárias. Por exemplo, você pode estar apto a iniciar um sistema de contabilidade em um disco de 8 polegadas ou em dois **minifloppies.** A maioria dos microcomputadores nacionais podem ser configurados para conexão de dois drives de **minifloppies;** alguns permitem configuração até quatro drives, porém em sua grande maioria não utilizam discos de 8 pelegadas.

Os dados nos discos flexíveis são gravados em trilhas concêntricas, que são divididas em setores. Quando um disco é inicializado ou "formatado", códigos são gravados nele para identificar cada trilha e setor, e também um diretório do seu conteúdo é gravado num setor e trilha específicos. O circuito do controlador de disco e o programa D.O.S. (Sistema Operacional em Disco) lêem o diretório para localizar onde o programa e arquivo de dados estão gravados no disco, para então moverem a cabeça de escrita/leitura diretamente para lá. Isto não só permite acesso mais rápido à informação no disco, como também o torna mais versátil.

Um sistema mínimo de disco pode ser usado com 16 KB de RAM, porém, pouco espaço estará disponível para os programas. Visto que uma das vantagens primárias de um sistema de disco é a capacidade de armazenamento de grandes programas e bastante quantidade de dados, recomendamos um mínimo de 32 KB de RAM; isto significa 16 KB para teclado e 16 KB para as interfaces de expansão.

O primeiro drive de disco é designado por **drive 0** (zero). Quando o sistema está em utilização o **drive 0** sempre contém o disquete com o programa D. O. S. (Sistema Operacional em Disco). Este disquete também contém um programa executivo para controle das operações e muitos programas auxiliares, incluindo o Basic.

No próximo número continuaremos apresentando os discos magnéticos. Até lá.

GLOSSÁRIO DE TERMOS TÉCNICOS

A rapidez com que ocorre o progresso da **informática** é tanta, que pessoas envolvidas ativamente no sistema de **comunicação** sentem dificuldades de se manterem atualizadas com o progresso fora de sua própria esfera particular. O povo em geral está, geralmente, e com razão, completamente confuso com o que está acontecendo. Como uma introdução para este **GLOSSÁRIO** é mais prático descrever brevemente algumas das técnicas envolvidas na nova tecnologia; isto ainda poderia ser usado para dar alguma idéia da extensão pretendida pelo mesmo.

Podemos começar por **COMPUTADORES,** já que eles representam um dos elementos básicos da nova tecnologia. Os computadores razoavelmente potentes vêm diminuindo rapidamente, tanto em tamanho quanto em custo, desde a década passada. Com o advento do microprocessador, os computadores evoluíram tanto que já podem ficar sobre uma mesa de escritório gerando e transmitindo informações. Um exemplo típico deste uso é a quantidade enorme de textos produzidos no mundo moderno por computadores especializados chamados **processadores de texto,** que podem produzir automaticamente centenas e milhares de cartas e documentos em questão de minutos. A saída desses computadores pode ser com a tradicional impressora ou pode ser transmitida com a mesma facilidade, através de outros canais de comunicação, a um terminal distante. Portanto, uma carta pode ser enviada através do **correio eletrônico** em vez de se usar o serviço postal tradicional.

Os métodos tanto para a "entrada" como para a "saída" de informações em computadores estão se diversificando com uma velocidade incrível. A maior parte da informação ainda é convertida para a forma eletrônica através de um "teclado", como numa máquina de escrever. Porém, já é possível conversar com um computador; utilizando-se algumas frases pode-se ter acesso direto **(entrada vocal)**. É esperado que as limitações atuais, como no caso do vocabulário, sejam superadas parcialmente, ainda na década de 80.

Esta pequena consideração da nova tecnologia está longe de ser compreendida, mas poderia servir para ilustrar sua diversidade. Quase inevitavelmente, as rápidas mudanças dentro do campo levam a um vocabulário que rapidamente muda. Por causa disso, muita confusão é criada não apenas porque os termos novos podem não ser muito fáceis de serem entendidos por todos os leitores, mas também porque o mesmo termo pode ser usado de diferentes formas. A proposição principal deste **GLOSSÁRIO** é ajudar a desfazer tanta confusão, mostrando os termos juntos e definindo os mais importantes e de uso mais corrente nas várias partes do campo da informática. O critério principal na sua elaboração foi a apresentação do termo técnico de forma que sugerisse uma ampla utilização. Assim sendo, conservamos em negrito o termo em inglês seguido pela tradução, quando significativa na língua portuguesa, entre parênteses, e finalmente a definição, por algumas vezes longa, objetivando um enfoque maior num contexto apropriado.

Poderíamos dizer que este **GLOSSÁRIO** é destinado a não técnicos, mas isto não seria verdadeiro pois existem técnicos de informática cuja familiarização com a nova terminologia é limitada; logo, o seu conteúdo objetiva os vários grupos de pessoas cujos interesses alcançam o campo da informática.

ABEND *(abnormal end)* — Fim irregular de uma tarefa de computador devido a erros ou intervenção do operador.

ABORT *(abortar)* — Utilizado em computação para abandonar uma atividade, geralmente porque um erro foi realizado.

ABSOLUTE ADDRESS *(endereço absoluto)* — Endereço real de uma localização de armazenamento, expressa em termos do sistema de numeração do código de máquina.

ACCESS *(Automated Catalog of Computer Equipment and Software Systems, US Army)* — Catálogo automático de equipamentos computadorizados e sistemas de software da Marinha dos Estados Unidos (US).

ACCESS ARM *(braço de acesso)* — Dispositivo que move o sistema de cabeças de leitura e gravação dos drives de disco ou fita magnética.

ACCESS MODE *(modo de acesso)* — Em COBOL é uma técnica utilizada para consultar um registro lógico, que faz parte de um arquivo designado para um dispositivo de armazenamento de massa.

ACCESS TIME *(tempo de acesso)* — É o tempo transcorrido entre o instante que um endereço é enviado à memória e o instante que os dados estão disponíveis para utilização. Uma vez que a memória pode ser acessada para leitura ou gravação (escrita) de dados, o tempo de acesso é especificado como a média entre os tempos de escrita e leitura de dados na memória.

ACCURACY *(exatidão)* — Livre de erro e/ou tamanho de um erro: usado em relação a programas, dados e operações de máquinas.

ACIA *(Asynchronous Communications Interface Adapter)* — Dispositivo que formata e controla os dados em um interface de comunicação assíncrona.

ACK *(ACK nowledgement)* — Sinal afirmativo de reconhecimento enviado numa linha de transmissão para indicar que um bloco de dados está sendo recebido ou que o receptor está pronto para receber dados.

ACL *(Audit Command Language)* — Linguagem de programação de alto nível.

ACOUSTIC COUPLER *(acoplador acústico)* — Dispositivo capaz de transmitir e receber sons numa linha telefônica, permitindo o acoplamento de um computador e um terminal, e utilizando um **modem** e um aparelho telefônico.

ACOUSTIC DELAY LINE — Retardo de tempo de propagação de ondas sonoras numa linha de transmissão.

ACTIVE FILE *(arquivo ativo)* — Um arquivo qualquer de um computador em uso corrente.

No próximo número continuaremos o nosso GLOSSÁRIO de termos técnicos. Até lá.

CEDM

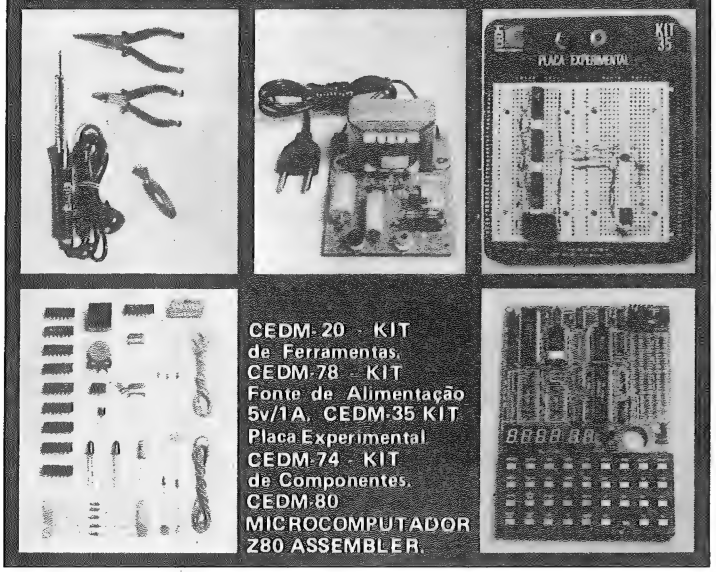
CURSOS DE APERFEIÇOAMENTO

MAIS SUCESSO PARA VOCÊ!

**Comece uma nova fase na sua vida profissional.
Os CURSOS CEDM levam até você o mais moderno ensino técnico programado e desenvolvido no País.**

CURSO DE ELETRÔNICA DIGITAL E MICROPROCESSADORES

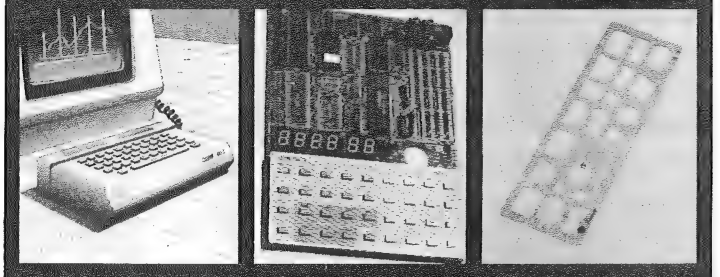
São mais de 140 apostilas com informações completas e sempre atualizadas. Tudo sobre os mais revolucionário CHIPS. E você recebe, além de uma sólida formação teórica, KITS elaborados para o seu desenvolvimento prático. Garanta agora o seu futuro.



CEDM-20 - KIT de Ferramentas. CEDM-78 - KIT Fonte de Alimentação 5v/1A. CEDM-35 KIT Placa Experimental CEDM-74 - KIT de Componentes. CEDM-80 MICROCOMPUTADOR Z80 ASSEMBLER.

CURSO DE PROGRAMAÇÃO EM BASIC

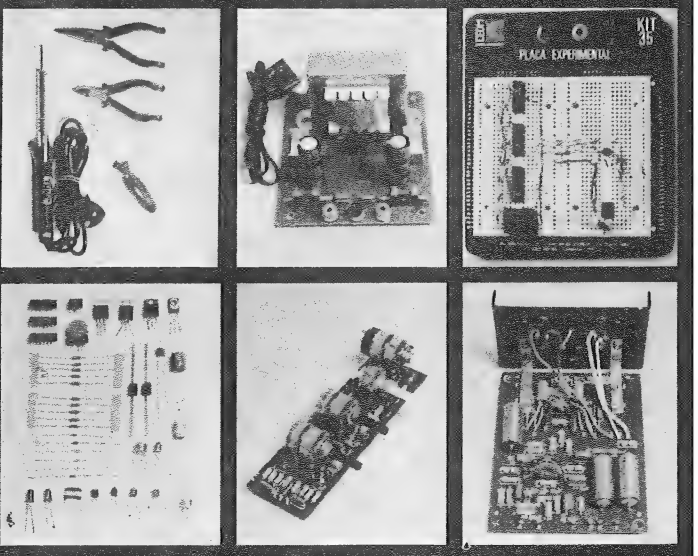
Este CURSO, especialmente programado, oferece os fundamentos de Linguagem de Programação que domina o universo dos microcomputadores. Dinâmico e abrangente, ensina desde o BASIC básico até o BASIC mais avançado, incluíndo noções básicas sobre Manipulação de Arquivos, Técnicas de Programação, Sistemas de Processamento de Dados, Teleprocessamento, Multiprogramação e Técnicas em Linguagem de Máquina, que proporcionam um grande conhecimento em toda a área de Processamento de Dados.



KIT CEDM Z80 BASIC Científico. KIT CEDM Z80 BASIC Simples. Gabarito de Fluxograma E-4. KIT CEDM SOFTWARE Fitas Cassete com Programas.

CURSO DE ELETRÔNICA E ÁUDIO

Métodos novos e inéditos de ensino garantem um aprendizado prático muito melhor. Em cada nova lição, apostilas ilustradas ensinam tudo sobre Amplificadores, Caixas Acústicas, Equalizadores, Toca-discos, Sintonizadores AM/FM, Gravadores e Toca-Fitas, Cápsulas e Fonocaptadores, Microfones, Sonorização, Instrumentação de Medidas em Áudio, Técnicas de Gravação e também de Reparação em Áudio.



CEDM-1 - KIT de Ferramentas. CEDM-2 - KIT Fonte de Alimentação + 15-15/1A. CEDM-3 - KIT Placa Experimental CEDM-4 - KIT de Componentes. CEDM-5 - KIT Pré-amplificador Estéreo. CEDM-6 - KIT Amplificador Estéreo 40w.

Você mesmo pode desenvolver um ritmo próprio de estudo. A linguagem simplificada dos CURSOS CEDM permite aprendizado fácil. E para esclarecer qualquer dúvida, o CEDM coloca à sua disposição uma equipe de professores sempre muito bem acessorada. Além disso, você recebe KITS preparados para os seus exercícios práticos.

Ágil, moderno e perfeitamente adequado à nossa realidade, os CURSOS CEDM por correspondência garantem condições ideais para o seu aperfeiçoamento profissional.

GRÁTIS

Você também pode ganhar um MICROCOMPUTADOR.

Telefone (0432) 23-9674 ou coloque hoje mesmo no Correio o cupom CEDM.

Em poucos dias você recebe nossos catálogos de apresentação.

CEDM

Avenida São Paulo, 718 - Fone (0432) 23-9674.
CAIXA POSTAL 1642 - CEP 86100 - Londrina - PR

CURSO DE APERFEIÇOAMENTO POR CORRESPONDÊNCIA

Solicito o mais rápido possível informações sem compromisso sobre o CURSO de . . . . . .

Nome. . . . . .

Rua. . . . . .

Cidade . . . . . .

Bairro. . . . . CEP . . . . . .

EDUCAÇÃO/TECNOLOGIA

Segue abaixo a publicação do trabalho do leitor Juarêz Fernandes Ramos, selecionado entre todos os que nos enviaram monografia sobre o tema "A Automação Bancária". O vencedor também recebe um micro-computador TK-82C, cortesia da Microdigital.

Juarêz Fernandes Ramos
Rua Coronel Monjardim, nº 99 – 12º andar.
Vitória – ES

A evolução tecnológica nos serviços e equipamentos de informática vem proporcionando o desenvolvimento de uma série de máquinas, especialmente projetadas para automatizar as operações de uma agência bancária.

As evidentes possibilidades de redução do custo operacional, associadas à viabilidade de oferta de novos serviços, vem fazendo com que os bancos invistam decisivamente nesta área, incentivando o desenvolvimento de diversos tipos de terminais, tanto OFF-LINE quanto ON-LINE, com estes últimos ligados ao computador central diretamente ou através de micro-computadores.

Para utilização direta pelos clientes, estão disponíveis terminais que executam funções como verificação de saldos e impressão de extratos. Estes terminais são acessados pela inserção de cartões magnéticos onde estão gravadas "senhas" livremente estabelecidas pelos usuários. A leitura automática destas senhas e a confrontação da mesma com a digitada pelo usuário garantem o sigilo das operações, tornando exclusivo para cada usuário o acesso à respectiva conta-corrente.

Ainda no contexto de automação bancária estão inseridos os equipamentos desenvolvidos para o Cartão Ouro Magnético, do Banco do Brasil. O sistema, em implantação inicial em Brasília, utiliza equipamentos especiais para gravar, nas agências, os limites de crédito dos cartões. Um segundo tipo de equipamentos, OFF-LINE, está sendo instalado em lojas, para efetuar automaticamente os depósitos referentes a compras ou saques em dinheiro, eliminando totalmente a necessidade de emissão de cheques.

Certamente o domínio já obtido desta tecnologia e as perspectivas de redução de custos dos equipamentos tenderão a acelerar o processo de automação das agências bancárias.

Como próximo estágio merece destaque o lançamento de máquinas que praticamente substituem uma agência bancária, como o Terminal de Caixa Automático Itaú e o ATM Bradesco. Estes equipamentos funcionarão 24 horas por dia, ao lado de agências (porém com acesso independente) ou em postos situados em locais públicos, estando capacitados para operações de depósitos, transferência de fundos, etc. As funções oferecidas serão acessadas pelos usuários através de cartões, processadas por um conjunto integrado de micro-processadores e "software" específico, ligados ao computador central ou das agências.

Ao contrário do que se esperava, o processo de automação dos serviços bancários não gerou desemprego. Mas limitou a criação de novas vagas, além de ter descaracterizado a profissão de bancário.

A tendência do setor financeiro a se expandir cada vez mais, apesar da recessão, tem feito com que vagas eliminadas pela automação sejam deslocadas para novas agências ou centros administrativos sem haver, contudo, novas contratações. O efeito negativo deste processo é deixar de fora uma massa de indivíduos que normalmente se deslocariam para o mercado de trabalho bancário, provocando desemprego potencial difícil de ser mensurado à primeira vista.

O Bradesco foi quem, pioneiramente, introduziu o uso de computadores na atividade bancária do país, no início da década de 60, e hoje possui um dos maiores, talvez o maior centro de Processamento de Dados da América Latina, localizado na Cidade de Deus, Osasco. Consta que os investimentos do banco, na área da informática, atingirão os Cr$ 25 bilhões até dezembro deste ano.

Como exemplo deste "mundo novo" em que os bancos do país mergulharam merece destaque o CMC-7, uma máquina que lê o cheque na "boca da máquina" fazendo numa única operação, realizada pelo próprio caixa que paga o cheque, todos os registros necessários a esta operação. Antes, a máquina existente era de "leitura constante", em que os cheques não podiam ser introduzidos manualmente. Com a criação da Digilab, pelo Bradesco, foi desenvolvida, então, esta máquina, que hoje se encontra pelo país, e bancos franceses e norte-americanos a utilizam com a denominação de E-13B.

Também para apressar o seu programa de automação, o Banco Itaú preferiu recorrer à criação de uma empresa própria para absorção de tecnologia importada e desenvolvimento e produção de equipamentos. Assim, em 1979 surgiu a Itaú Tecnologia S/A (Itautec), que até 1982 resumia suas atividades à produção de "Hardware" e "Software". Através dela, o Itaú já conta com 160 agências eletrônicas interligadas, envolvendo mais de 3 mil terminais e aproximadamente 1 milhão de transações diárias. Sendo a primeira e uma das maiores redes de "banco eletrônico" implantadas no país, o Itaú continua colocando novos produtos no mercado. Perde para o Bradesco na adoção da ATM, a qual denomina de Caixa Eletrônica Itaú, que para desenvolver e produzir aqui importou tecnologia externa através da Itautec. É uma máquina que aceita retiradas ON-LINA e depósitos OFF-LINA, além de fornecer aos clientes informações diversas a qualquer hora, durante os sete dias da semana.

Constata-se então, que esta guerra deflagrada entre os poderosos bancos tende a trazer benefícios para a população em geral, seja pela agilização de operações e redução de custos, seja pelo investimento maciço em desenvolvimento de tecnologia com a conseqüente geração de novos empregos.
Que venha então a automação de todas as agências de todos os bancos, com o conseqüente domínio de uma tecnologia até então importada a peso de ouro. E que então se exporte esta mesma tecnologia, agora Tupiniquim, a outros países, como aliás já vem ocorrendo com a CMC-7, mais conhecida internacionalmente como E-13B; pois é disto que esta pobre terra do Pau Brasil precisa.

LOGUS III

A TECNOLOGIA BRASILEIRA

- CPU com 64K a 16 Megabytes de RAM
- 8 canais de comunicações
- Até 32 discos fixos ou removíveis
- Até 15 programas ativos em multi-programação
- Sistema operacional baseado em mapeamento (páginas de 4 Kbytes cada) sem "wait state"

LOGUS COMPUTADORES LTDA
Rua Dr. Freire, 97
Fone (011) 270-3803
CEP 03101 - Mooca - SP

MEMÓRIAS

Por Cesar da Costa

3ª Parte

MEMÓRIAS SEMICONDUTORAS

Como já vimos anteriormente, os flip-flops podem ser considerados como células básicas de memória, e reunidos formam unidades fundamentais de armazenamento de informação. Com o desenvolvimento das técnicas de integração em larga escala (LSI, VLSI) uma grande variedade de memórias foram e estão sendo implementadas. As memórias semicondutoras apresentam algumas vantagens sobre as memórias de núcleo magnético, tais como *baixo custo por bit, menores dimensões, menor tempo de acesso* **e** *leitura não destrutiva.* **Por outro lado, necessitam de alimentação do circuito externo para manterem a informação armazenada (memórias voláteis).**

O primeiro tipo de memória que vamos abordar são as memórias RAM (RAMDOMIC ACESS MEMORY), também conhecidas como *memórias de acesso aleatório de escrita e leitura.* Nesse tipo de memória a informação pode ser lida ou escrita aleatoriamente, isto é, pode-se ter acesso a qualquer endereço que se deseje. As memórias RAM são construídas com tecnologia bipolar ou CMOS. Essas memórias são normalmente utilizadas para armazenamento de dados que estão sendo utilizados pelo sistema.

As memórias RAM's bipolares são de pequena capacidade de armazenamento e de alta velocidade de operação; os projetistas devem observar com cuidado os tempos de *acesso* e de *ciclo,* pois tornam-se críticos em sistemas de alta velocidade.

Um *chip* (circuito integrado) de memória é constituído de 2^n palavras por *K* bits, sendo necessário *n* bits para a formação do código de endereço, *K* bits de entrada de dados, *K* bits de saída de dados, entradas de comando de escrita/leitura e seleção. Por exemplo, o CI 74200 é uma RAM organizada em 256 x 1, ou seja, 256 palavras de 1 bit. O CI 7489 é outro exemplo de RAM organizada em 16 x 4, ou seja, 16 palavras de 4 bits. O CI 3601 é uma PROM organizada em 256 x 4, isto é, 256 palavras de 4 bits.

A capacidade de memória de um chip é normalmente dimensionada pela quantidade de palavras que ele pode armazenar, e não pelo número de bits. Por exemplo, um chip de memória com uma organização de 256 x 4 tem capacidade para 256 palavras de 4 bits. Sua capacidade de 1K bit é real, porém não tem grande importância pois num sistema digital o mais importante é o tamanho da palavra digital que será armazenada, e não o bit individualmente. Assim sendo, quando nos referimos a um sistema de 1 KB (Byte) de capacidade, devemos ter em mente que o sistema está organizado em 1024 (1K) palavras de 8 bits (1 Byte), ou 1K x 8, ou ainda 1024 x 8.

EXPANSÃO DE MEMÓRIA

A expansão de um sistema de memória pode consistir em:

- Expansão do tamanho da palavra de memória — consiste no aumento do número de bits da palavra, não alterando o número de endereços do sistema de memória. Por exemplo, um sistema de memória de 4 bits pode ser expandido para 8, 16 ou 32 bits.
- Expansão do número de palavras da memória — consiste no aumento do número de palavras do sistema de memória, alterando o número de endereços do sistema de memória. Por exemplo, um sistema de 256 palavras por 4 bits pode ser expandido para 1024 palavras por 4 bits, ou ainda, um sistema de 16 KB pode ser expandido para 48 KB.

A implementação prática de uma expansão do tamanho da palavra de memória constitui-se no fato de se dese-

jar uma palavra de memória com número de bits maior do que aquele fornecido pelo *chip* a ser utilizado, o que pode ser conseguido simplesmente pela colocação em paralelo das entradas e saídas e conexão das linhas de endereços de mesmo índice.

O número de circuitos integrados que determinarão a palavra desejada é dado pela seguinte fórmula:

$$\text{NÚMERO DE CI} = \frac{\text{Nº de bits da palavra desejada}}{\text{Nº de bits da palavra do CI}}$$

Na implementação da expansão do número de palavras o problema não é muito simples, pois o projetista formará um sistema com um número de palavras e endereços maior que aqueles fornecidos pelo *chip*. Além da conexão das entradas, saídas e linhas de endereço de mesmo índice, o projetista deverá implementar um circuito de seleção e decodificação de endereço que se tornará a parte mais difícil do projeto.

O número de circuitos integrados que determinarão o novo número de palavras do sistema será dado pela fórmula:

$$\text{NÚMERO DE CI} = \frac{\text{Nº de palavras necessárias}}{\text{Nº de palavras do CI}}$$

Por exemplo, vamos projetar um sistema de memória com organização de 2K x 4 bits, utilizando o CI 74200.

Solução:

O CI 74200 é uma RAM organizada em 256 x 1 bit. Assim sendo, teremos que implementar uma expansão do tamanho da palavra e uma expansão do número de palavras do CI.

Número de *chips* que serão necessários para uma palavra de 4 bits:

$$\text{Nº de CI} = \frac{\text{Nº de bits da palavra desejada}}{\text{Nº de bits da palavra do CI}} = \frac{4}{1} = 4$$

Número de *chips* que serão necessários para 2048 (2K) palavras:

$$\text{Nº de CI} = \frac{\text{Nº de palavras necessárias}}{\text{Nº de palavras do CI}} = \frac{2048}{256} = 8$$

O sistema de memória solução é uma matriz que requer um arranjo de 8 linhas de chip por 4 colunas de chip, figura 1, num total de 32 chips.

O número de linhas de endereço será $2^n = 2048$, onde n é o número de linhas de endereço, logo $n = 11$.

Cada CI 74200 possui 8 linhas de endereço, logo, teremos que utilizar mais 3 linhas de endereço.

Todas as linhas de entradas de dados (DI) e saídas (DO) de mesmo índice são interconectadas em cada coluna.

Todas as 8 linhas de endereço de mesmo índice são interconectadas nos 32 chips.

Lembramos ao leitor a importância do projeto prático do circuito de endereçamento do sistema de memória que, embora fugindo ao tema abordado neste artigo, será apresentado como ilustração para o problema proposto.

Para o projeto do decodificador de endereço do sistema é fundamental que o projetista conheça todas as características elétricas do chip de memória. Por exemplo, o CI 74200 possui três linhas de controle ME que deverão ser nível baixo (zero) para a seleção do chip. Dessa maneira, essas três entradas serão utilizadas para a seleção da linha, sendo a linha 0 os endereços mais baixos de 0 a 255 e a linha 7 os endereços mais altos de 1792 a 2047, conforme tabela 1.

TABELA 1

CÓDIGO DA LINHA	NÚMERO DA LINHA	ENDEREÇO
0 0 0	Linha 0	0 – 255
0 0 1	Linha 1	256 – 511
0 1 0	Linha 2	512 – 767
0 1 1	Linha 3	768 – 1023
1 0 0	Linha 4	1024 – 1279
1 0 1	Linha 5	1280 – 1535
1 1 0	Linha 6	1536 – 1791
1 1 1	Linha 7	1792 – 2047

O problema se resume num simples projeto de álgebra de boole no qual para três entradas serão necessárias 24 saídas divididas em 8 grupos de 3, e que serão verdadeiras quando somente um grupo tiver todas as linhas *nível baixo.*

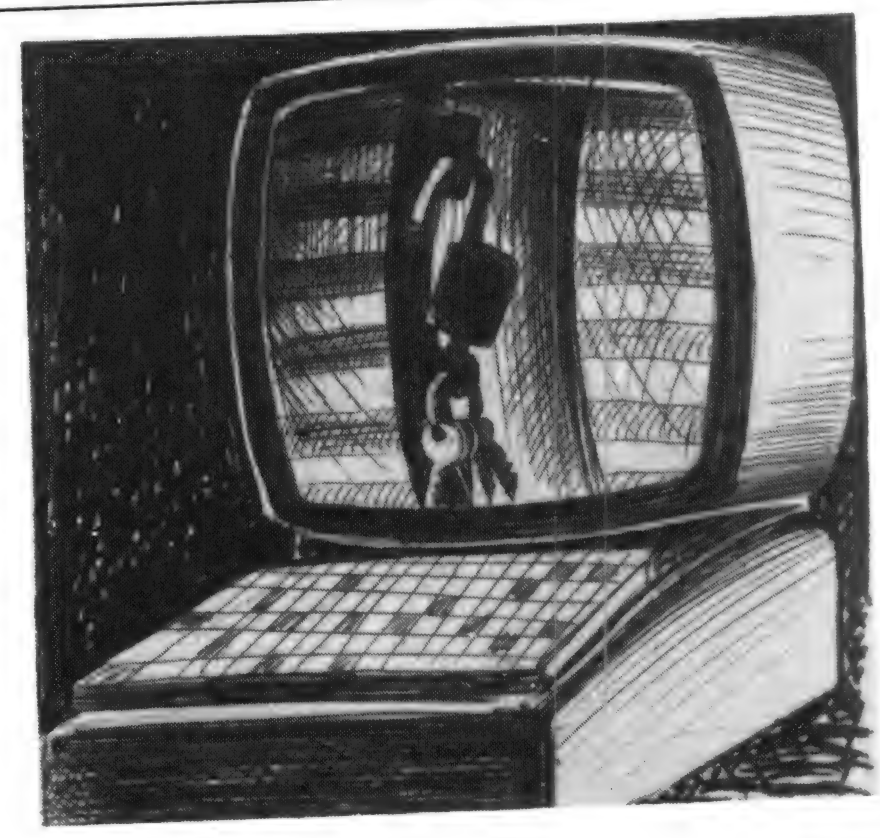
Os oito bits menos significativos das onze linhas de endereço vão para as oito entradas de endereço dos 32 chips. Os três bits mais significativos e seus complementos vão para as entradas *ME* dos quatro chips componentes de cada linha. Somente a linha selecionada terá as três entradas *ME* nível baixo, ativando a palavra de 4 bits.

Aconselhamos ao leitor simular vários endereços na figura 1 e observar o grupo de CIS que será selecionado. Assim, poderá verificar que no barramento dos onze bits de endereço, os oito bits menos significativos (LSB) endereçam a palavra contida no CI e os três mais significativos (MSB) endereçam a linha de memória. Por exemplo, o endereço 1023:

A0	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10
1	1	1	1	1	1	1	1	1	1	0
LSB – Palavra 512								MSB – 3 linhas		

ERRATA

Na 2ª parte, revista nº 9, página 45, na figura nº 3, onde está escrito "VARIAÇÃO DO FLUXO + BM PARA -BR leia-se "VARIAÇÃO DO FLUXO DE +BR PARA - BM".



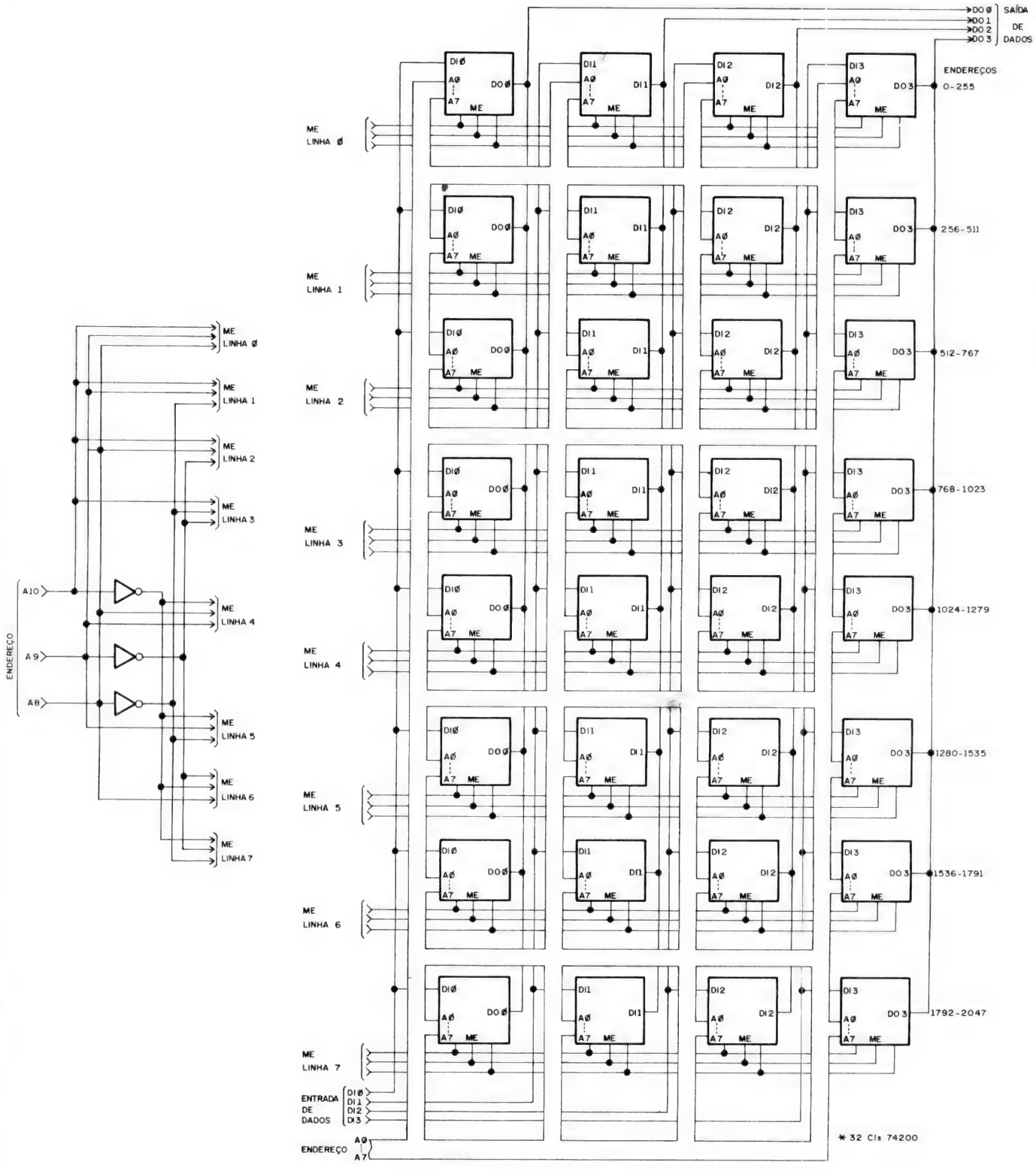


FIGURA 1 – SISTEMA DE MEMÓRIA – ORGANIZAÇÃO 2K x 4 BITS E CIRCUITO DE ENDEREÇAMENTO

NO PRÓXIMO NÚMERO APRESENTAREMOS AS "MEMÓRIAS SEMICONDUTORAS CMOS".

Cartas

ECO E RESSONÂNCIA

Foi com enorme prazer que ao tomar conhecimento da revista INTERFACE, através de seu exemplar nº 3, solicitei uma assinatura da mesma. Este prazer vem sendo renovado a cada número que recebo e mais ainda quando encontrei a reedição dos exemplares já esgotados, e no nº 1 o "grito" (assim o vi) do Msc Tarcísio Neves da Cunha, com o qual faço "eco": "É mister tomar para nós um mercado que é nosso, sob pena de nos encontrarmos, a médio prazo, como meros operadores de pacotes fechados, adaptando a eles as nossas condições e não o contrário, que é a saudável adaptação da tecnologia às circunstâncias de cada caso".

Saudações ao Msc Tarcísio Neves da Cunha e a todos que fazem INTERFACE.

Emanuel B. Brito
Recife – PE

Agradecemos em nome do Tarcísio e de toda a nossa equipe por tudo o que você diz. Parece que alcançamos a "ressonância" desejada, Emanuel.

NÚMEROS ATRASADOS

Estou deveras satisfeito com toda a atenção recebida de vocês. Como vejo, não me arrependo de ter assinado esta revista da qual, novamente repito, me considero assíduo leitor.

Os exemplares que não tive oportunidade de adquirir são os de nº 01, 02 e 03, e envio-lhes cheque pagável para que vocês possam reservar esses três primeiros números que são imprescindíveis para mim.

Fico satisfeito em saber que começo a minha carreira na informática junto com o apoio de um instrumento valioso que é a revista INTERFACE.

Alfredo Luciano
São Bernardo do Campo – SP

Tomei conhecimento do lançamento da revista INTERFACE após já ter sido publicado alguns números, devido a mesma não ser encontrada nas principais bancas aqui em Volta Redonda a não ser por encomenda.

Analisando alguns números de INTERFACE já publicados constatei ser a mesma de grande qualidade pela versatilidade e utilidade dos artigos publicados principalmente para nós que trabalhamos com micros e utilizamos o versátil CPU Z-80 e interfaces.

Pelas razões expostas, desejo ser o novo assinante de INTERFACE, mas gostaria de saber se há possibilidade de após concretizada a assinatura os números já publicados serem incluídos, isto é, a partir do número 1.

Termino enviando meus cumprimentos pela qualidade da revista, agradeço a atenção e aguardo resposta.

Cláudio Luiz de Sá
Volta Redonda – RJ

Vocês já devem ter recebido orientações precisas sobre a aquisição de números atrasados de INTERFACE. Recebam nossos agradecimentos.

JOGO DE XADREZ EM BASIC

Antes de mais dizeres, gostaria de felicitá-los pela excelente publicação de informática que é a revista INTERFACE.

Gostaria de solicitar aos senhores, se for possível, a publicação de um jogo de xadrez (em Basic) ou informações que me levassem a adquirir o jogo supracitado. De preferência, compatível com o CP500 (Prológica).

Desde já agradeço a atenção e desejo que esta publicação prospere o máximo.

Paulo Cesar Bonetti
Juiz de Fora – MG

Seu pedido de programa de jogo de xadrez foi anotado e poderá ser atendido através do Suplemento Especial para o usuário, que iniciamos neste número.

Quanto a adquirir o jogo já pronto podemos adiantar que a Microsoft, fabricante de software, possui este jogo disponível. Não sabemos, no entanto, da existência ou não de representante ou revendedor aí em Juiz de Fora. Para maiores informações contate a própria Microsoft (nosso anunciante).

INTERCÂMBIO

Tenho 14 anos, sou estudante e possuo um micro DGT-100. Coleciono a revista INTERFACE desde o número um. Gosto da forma com que são debatidos os fatos e o uso de uma linguagem fácil e objetiva, tanto para os técnicos como para os iniciantes, o que não é o meu caso já que possuo um bom conhecimento sobre a linguagem Basic e alguns aspectos sobre o assembler.

Devido a minha pouca idade, a parte da revista que mais me interessa é justamente a dos programas que aparecem na coluna "Clube do Microsoftware". Por isso, resolvi escrever esta carta e com ela mandar um programa. Este programa simula um verdadeiro caleidoscópio, só que quem irá fazer os desenhos é o computador. Com o término de um desenho, outro será prontamente elaborado após pressionar **return**.

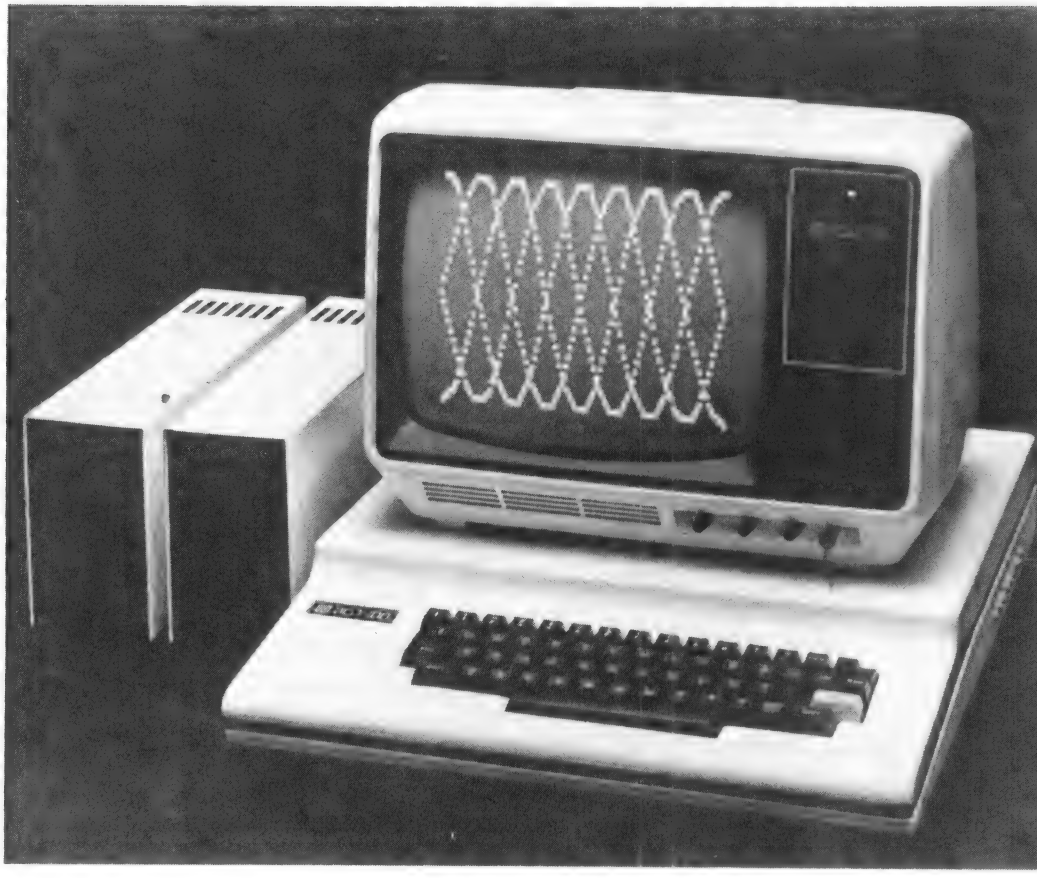
Desejaria que este anúncio também fosse publicado: gostaria de trocar informações e programas (principalmente jogos) com pessoas que possuam um microcomputador compatível ao TRS-80 modelos I ou II, de preferência na Ilha do Governador ou Ipanema.

Termino, fazendo votos que a INTERFACE progrida e que abra cada vez mais os campos da informática no Brasil.

André Luís N. Vianna
Rio de Janeiro – RJ

Seu programa vai ser publicado no nosso suplemento especial, que dedicamos a leitores como você; em breve o verá. E vá em frente, pois o caminho está aí. Progrida junto com INTERFACE.

DGP/M



TRABALHE COM O GRANDE SISTEMA DIGITUS

Lançado recentemente, o DGT-101 vem adquirindo o espaço merecido devido a sua eficiência nas grandes empresas.

O DGT-101 possui o sistema operacional DGP/M, totalmente compatível com o CP/M, sua memória é ampliada para 64k bytes em RAM.

Devido a formatação do DGP/M ser de 212 bytes por setor, a capacidade de armazenamento de sua unidade de drive é de 200K bytes sendo que, o sistema completo pode ter acesso a 800k bytes de memória.

Com o DGP/M você terá acesso à linguagens poderosas para aplicações comerciais e científicas, como: cobol, fortran, pascal, PL1, CBASIC que rodarão compiladas, fazendo com que o sistema tenha uma grande velocidade.

Por ser de grande eficiência e de interação simples pelo usuário, o DGP/M é amplamente utilizado no mundo dos computadores, razão pela qual a maioria dos software existentes são desenvolvidos sob este sistema operacional.

O sistema DGT-101 constitui-se do microcomputador com 64k bytes, interface para drive, duas unidades de disc-drive, interface paralela para impressora, vídeo verde, cabos necessários para ligação, manuais e discos.

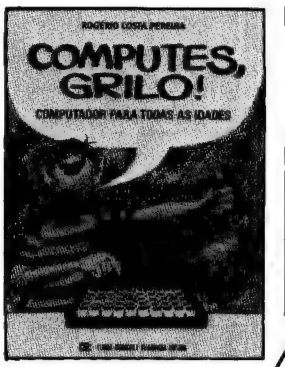
Como todos periféricos DIGITUS, a interface do sistema operacional DGP/M é totalmente compatível com o DGT-100, ou seja com o acréscimo da interface DGP/M você transforma seu DGT-100 em um DGT-101.

DIGITUS

DIGITUS - Ind. Com. Serv. de Eletrônica Ltda.
Rua Gávea, 150 - Tel.: (031) 332-8300 B Hte

• Aracaju (079) 224.1310/224.6111 • Brasília (061) 242.6344/226.8701 - 226.9201 - 224.2777/226.5006 - 225.4534 - 224.3505/226.2374/248.5030/561.3307 -248.6321 • Belo Horizonte (231) 222.7889 - 223.6947 - 226.6336 - 226.5734 - 226.9078 - 225.3305 - 225.2469 • Belém (091) 223.1090 - 224.9988 • Campinas (0192) 32.4155 - 32.3810/32.4445 • Campo Grande (067) 383.1068 - 382.6487 • Cuiabá (065) 322.9713/321.7929 • Curitiba (041) 232.1750 - 243.1731 • Fortaleza (085) 231.4822 - 227.5878 - 224.0544/244.4691 - 226.4922 - 224.7864 - 231.4910/231.4822/231.4001 • Florianópolis (0482) 23.1039 • Frederico Westphalen (055) 334.1550/334.1672 • Goiânia (062) 225.0022 - 224.0557 - 225.8598 - 223.1122 • João Pessoa (083) 221.6743 • Maceió (082) 223.3979 • Niterói (021) 714.0112 - 722.6791/717.1570 • Novo Hamburgo (0512) 93.4721 • Natal (084) 222.3212 • Montes Clarros (038) 221.8212 • Ouro Preto (031) 551.1933 • Poços de Caldas (035) 721.5810 • Porto Alegre (0512) 22.9782 - 26.8468 - 26.8246 - 40.1998 - 21.4189 • Recife (081) 326.9969 - 326.9318 - 222.4714 • Ribeirão Preto (016) 636.0586 • Rio de Janeiro (021) 264.0143 - 262.8737 - 322.4166 - 252.2752 - 221.8282/295.8194/267.8291/247.1339 - 252.2050/252.4080 -228.0734/248.8159/284.5649 - 247.7842 - 222.6088 - 259.1516 - 288.2650 - 267.1093/267.1443 - 252.9057 - 264.5784 - 263.1241 - 392.8965 - 286.4849 -591.3297/249.3166 • Salvador (071) 247.4936/ 245.6198 - 243.2684/242.9394 - 248.6666 - 235.4184 • Santa Maria (055) 221.7120 • São Paulo (011) 271.1215/544.5001 - 222.1511 - 283.0596 - 852.2958 -282.2105 258.3954 - 227.6100/227.4433 - 280.2322 - 258.4411 - 212.9004/210.0187 -61.4049/61.0949/881.0200/881.1156 • Taubaté (0122) 32.9807.

LIVROS



PEREIRA, Rogério Costa – "COMPUTES, GRILO! COMPUTADOR PARA TODAS AS IDADES", LTC – Livros Técnicos e Científicos Editora S.A.

Este livro foi desenvolvido visando os jovens acima de 10 anos, sem conhecimento anterior sobre computadores, que já tenham completado ao menos a quarta série do primeiro grau. Porém, pode se usado com proveito por qualquer pessoa que queira desenvolver uma boa noção do que é e para que serve o computador.

Seu objetivo principal é dar ao leitor uma base, ao mesmo tempo conceitual e prática, que estimule de forma explícita o desenvolvimento de seu "raciocínio lógico" e facilite, no futuro, sua inevitável interação com os computadores. A linguagem Basic é adotada, tendo em vista a sua simplicidade e ampla disseminação.

O livro está dividido em duas partes: a primeira é indispensável ao entendimento do que é o computador e para que ele serve; é de fundamental importância para a compreensão da segunda parte, sendo dividida em cinco módulos com assuntos diferentes e um sexto módulo, de revisão. Já nesta primeira parte, a partir do módulo 2, o leitor estará usando o computador. A segunda parte mostra como o computador é usado na prática, sendo dividida em cinco módulos.

O livro apresenta de forma simples e objetiva as técnicas de programação, permitindo que se desenvolva um programa desde a viabilidade até o levantamento do problema, a diagramação, a codificação e os testes.

SOFTWARE
O Combustível do Computador

Programas para CP 500, DGT 100, TRS 80, Dismac, Polymax e outros.
Temos disponíveis mais de 50 programas específicos, em fita ou diskete, para aplicações diversas.

TABELA DE PREÇOS – PROMOÇÃO ESPECIAL		
PROGRAMA	FITA (Cr$)	DISKETE (Cr$)
Contabilidade	21.120,00	126.736,00
Folha de Pagamento	–	50.688,00
Controle de Estoque	33.792,00	63.360,00
Editor de Textos	16.896,00	42.240,00
Mala Direta	25.344,00	63.360,00
Arquivo de Processos	12.672,00	–
Contas Pagar/Receber	25.344,00	63.360,00
Controle de Livros	8.448,00	–
Controle de Cheques	8.448,00	–
Biorritmo	8.448,00	12.672,00
Decisão	8.448,00	12.672,00
Histograma	8.448,00	12.672,00
Estatística	9.292,00	13.516,00
Gráficos	9.292,00	13.516,00
Obstáculo	9.292,00	13.516,00
Eq. Diferencial	9.292,00	13.516,00
Controle Saldo Bancário	–	42.240,00
Crediário	–	63.360,00
Multifunção	–	84.480,00

☐ CURSOS DE BASIC
☐ SOLICITE NOSSO CATÁLOGO OU VISITE NOSSO SHOW-ROOM.
☐ DESCONTOS PARA REVENDA
☐ ATENDIMENTO POR REEMBOLSO POSTAL PARA TODO O BRASIL.

CONTROLE DE ESTOQUE PARA TK-82
NEZ 8.000, CP 200 e TK-85
Preço de Lançamento: Cr$ 6.800,00

nasajon sistemas
Av. Rio Branco, 45 gr. 1311
Tel (021) 263-1241
CEP 20090 – Rio de Janeiro.

Você também encontra esses programas em nossos revendedores credenciados.

HA

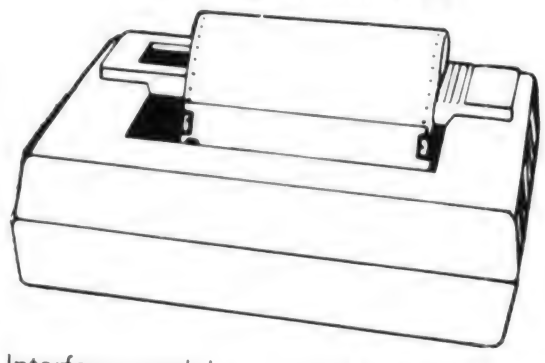
AS IMPRESSORAS QUE VÃO BEM COM TODOS OS MICROS.

GARANTIMOS O MENOR PREÇO DO MERCADO

Seja qual for o seu mini ou microcomputador, a Clappy tem a melhor opção em impressoras matriciais, pelo menor preço.
Para interfaces paralela ou serial, linhas de 80 ou 132 posições e velocidade de até 340 caracteres por segundo, a Clappy tem a impressora certa para a sua necessidade.
Aproveite para conhecer também o nosso lançamento na área de Plotters.

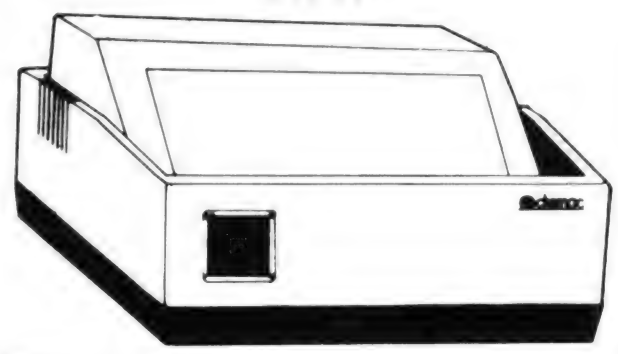
TESTADAS E OPERANDO COM TODOS OS MICROS

ELEBRA
EMILIA 8010/8030



Interface paralela ou serial (75 a 9600 bps), 100 cps, 80 ou 132 posições, matriz de pontos 9 x 7, densidade horizontal 5 ou 10 cpi, densidade vertical 6 ou 8 lpi, formulário contínuo 1 a 5 vias.

DISMAC
STAR



Interface paralela, 80 cps, 80 posições, matriz de pontos 9 x 7, densidade horizontal 5 a 16,5 cpi, densidade vertical 4,6 ou 8 lpi, conjunto de caracteres gráficos, formulário contínuo 1 a 4 vias ou papel carta.

POLYMAX
POLYPRINT

Interface paralela ou serial (110 a 1200 bps), 90 cps, 80 posições, matriz de pontos 7 x 9, densidade horizontal 5,2 a 17,1 cpi, densidade vertical 6 lpi, formulário contínuo 1 a 4 vias ou papel carta.

PROLÓGICA
P 720

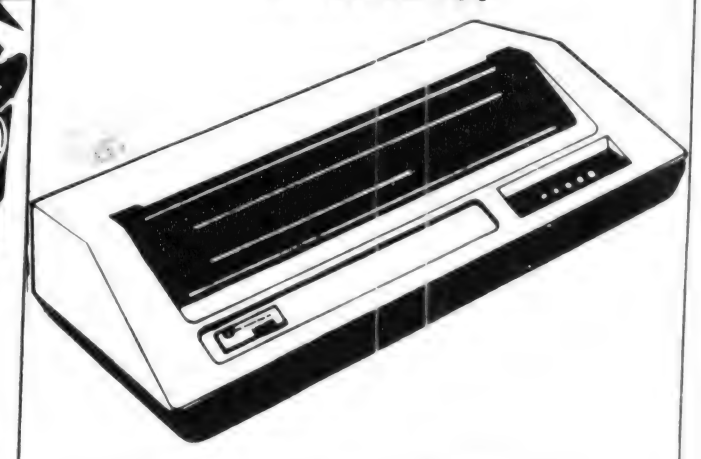
Interface paralela e serial, 200 cps, 132 posições, matriz de pontos 7 x 9, densidade horizontal 10 cpi, densidade vertical 6 lpi, formulário contínuo de 1 a 6 vias.

ELGIN
ELGIN LADY

A PRIMEIRA BRASILEIRA COM RESOLUÇÃO GRÁFICA

Interface paralela ou serial, 100 cps, 132 posições, matriz de pontos 7 x 9, densidade horizontal 10 a 16,6 cpi, densidade vertical 6 ou 8 lpi, conjunto de caracteres gráficos, formulário contínuo 1 a 6 vias ou papel carta.

GLOBUS
M 100 80/100



Interface paralela e serial (75 a 9600 bps), 100 cps, 80 ou 132 posições matriz de pontos 5 x 7, densidade horizontal 8 a 16,5 cpi, densidade vertical 6 ou 8 lpi, caractere reverso, negrito, formulário contínuo 1 a 4 vias.

STRATUS
S 310

Interface paralela ou serial (110 a 9600 bps), 112 cps, 80 posições, matriz de pontos 7 x 9, densidade horizontal 5,2 a 17,1 cpi, densidade vertical 6 lpi, caracteres sublinhados, formulário contínuo 1 a 4 vias ou papel carta.

BAUSCH & LOMB
PLOTTER DMP - 29

PRONTA ENTREGA

Traçador gráfico digital, resolução de 0,005 pol., 8 penas, mudança automática de pena, velocidade axial de 16 pol/seg., diagonal de 22 pol/seg., interface RS 232 C, 93 caracteres maiúsculos e minúsculos, em 255 tamanhos, 360 ang., rotação, área de traçado de 10 x 15 pol. Opera com qualquer microcomputador com interface serial (inclusive AP II e CP 500).

LANÇAMENTO EXCLUSIVO

INTERFACES/ACESSÓRIOS/ SUPRIMENTOS

INTERFACES PARALELA E SERIAL PARA AP II, DGT-100, CP 500, MAXXI E MICROENGENHO
CABOS DE LIGAÇÃO
FITAS PARA IMPRESSORA
FORMULÁRIO CONTÍNUO 80 E 132 POSIÇÕES EM CAIXAS DE 1 OU 3 MILHEIROS.

A CLAPPY DÁ GARANTIA E ASSISTÊNCIA TÉCNICA

Venha à nossa loja ou solicite a visita de um representante

Clappy

Centro: Av. Rio Branco, 12 - loja e sobreloja
234-0214 • - 253-3170 • 253-3395 • 283-3588 • 234-1015
Rio de Janeiro - RJ

Entregamos em todo o Brasil pelo reembolso VARIG

Breve Clappy Copacabana: Show Room: Rua Pompeu Loureiro, 99

Por T. Munnecke, R. F. Walters, J. Bowie, C. B. Lazarus e D. A. Bridger.

CARACTERÍSTICAS E COMPARAÇÕES COM OUTROS SISTEMAS DE PROGRAMAÇÃO

3ª Parte

MUMPS

O MUMPS foi originalmente projetado para os minicomputadores, explorando a natureza dedicada dos pequenos computadores. Utiliza o recurso mais barato (tempo de processador central) e diminui o recurso mais caro (tempo de pessoal).

SISTEMAS DE BANCO DE DADOS

O aspecto de MUMPS mais importante, provavelmente, não é qualquer característica específica ou conjunto de características, mas sim a integração de uma linguagem interativa com um sistema de banco de dados partilhado. Para o programador MUMPS, o banco de dados se assemelha a uma grande matriz de dados estruturada em árvore. Ele é capaz de referenciar esses dados de uma maneira simples e direta. As preocupações sobre a estruturação física e a alocação de espaço são deixadas para o sistema operacional MUMPS. Esse fator, mais do que qualquer outro, é responsável pela rapidez com que as aplicações de MUMPS podem ser projetadas e desenvolvidas.

A motivação original para esse projeto de banco de dados era facilitar o acesso on-line a um banco de dados partilhado e dinâmico.

Quando o MUMPS foi inicialmente desenvolvido havia poucas ferramentas para alcançar este objetivo e certamente não era capaz de ser executado em computadores de pequeno a médio porte. Desde então, numerosos sistemas de gerenciamento de dados foram projetados para esse fim. Alguns desses sistemas operam como sistemas de recuperação de informação geral de forma *stand-alone;* outros são projetados para aumentar uma linguagem geral de programação, tais como COBOL, FORTRAN ou PL/1; e outros, ainda, têm ambas as capacidades. É o *approach* de ampliações de uma linguagem geral que é relevante como uma alternativa para o MUMPS, e que será agora discutido.

Há muitos níveis de capacidade de gerenciamento de dados. O nível mais alto é geralmente chamado de *sistema de gerenciamento de banco de dados.* Esses sistemas originam-se de programas utilitários gerais que foram escritos para executar certas manipulações comuns em qualquer coleção de dados, daí evitando-se essencialmente a programação redundante. Um sistema completo de gerenciamento de banco de dados inclui um recurso para a definição central de todos os dados e maioria dos inter-relacionamentos de dados, permitindo aos programas de aplicação solicitarem a recuperação de dados e operações de armazenamento de uma forma lógica e de alto nível. Tal sistema também tenta assegurar a privacidade e integridade de seus dados.

Existem níveis menores de gerenciamento de dados que colocam mais responsabilidade nos programas de aplicação. O gerenciamento de registros, por exemplo, auxilia no

armazenamento e recuperação de registros "nomeados", mas podem não ter conhecimento algum das relações entre os registros ou da estrutura dentro dos registros. Este relatório examinará as características mais relevantes desses tipos diferentes de gerenciamento de dados e tentará determinar onde o MUMPS fica no espectro das capacidades.

GERÊNCIA DE REGISTROS

Quase todos os sistemas de gerência de dados, o MUMPS fica excluído, giram em torno do conceito de um *registro*. Um registro é uma coleção de dados que pode ser recuperado, arquivado ou copiado como uma única unidade. Dentro do registro existem itens também chamados de campos que podem ser examinados separadamente e processados, uma vez que o registro tiver sido recuperado, isto é, localizado no banco de dados e copiado na área de trabalho. Quando um programa tiver completado o processamento de um registro na área de trabalho ele pode copiar o registro para o banco de dados, dessa forma atualizando simultaneamente todos os itens alterados. Um novo registro é montado na área de trabalho e copiado de maneira similar para acrescentá-lo ao banco de dados.

Os métodos de acesso ao arquivo são técnicas pelas quais os registros individuais podem ser armazenados e recuperados. Eles são os recursos fornecidos pelo sistema operacional, aos quais um programa de aplicação pode chamar. Os métodos de acesso mais comuns são:

Seqüencial — **Os registros podem somente ser recuperados ou gravados em seqüência a partir do começo até o fim do arquivo. Este método de acesso é necessário quando um arquivo é armazenado em fita magnética, mas é também usado para arquivos em disco que necessitam somente ser manipulados em seqüência, assim como um arquivo que contém as transações do dia em ordem de ocorrência.**

Randômico ou Direto — **Os registros podem ser referenciados por uma chave numérica que dá o endereço físico ou da qual um endereço físico pode ser calculado. Esta técnica é inadequada se as chaves forem esparsas, já que uma posição física deve estar disponível para cada possível chave, e exige que todos os registros em um arquivo sejam de tamanho fixo. O problema de chave esparsa pode ser aliviado pelas técnicas de *hashing* e tratamento de colisões, mas uma distribuição pobre de chaves pode destruir tais técnicas.**

Indexado — **Cada registro é referenciado por uma chave numérica ou, mais comumente, alfanumérica, que é geralmente um dos itens de dados do registro. O sistema operacional mantém um índice que ele pesquisa para encontrar o endereço físico do registro. Há um grande número de possíveis métodos para construir e pesquisar tal índice, mas o mais comum é manter um índice hierárquico ordenado. Algumas vezes o sistema operacional tenta colocar todos os registros com chaves muito relacionadas no mesmo cilindro, para facilitar o acesso seqüencial. Freqüentemente a estrutura de índice hierárquico consiste de um índice de cilindro para todos os registros no mesmo cilindro e um índice de volume que é usado para encontrar o cilindro correto. Isto pode ser muito rápido se o arquivo não mudar muito. As chaves geralmente têm que ser de tamanho fixo.**

Indexado Hierarquicamente — **Este é parecido com o indexado, mas permite que um registro tenha várias chaves que sejam conceitualmente concatenadas e indexadas. O *DL/1* da IBM e os *INFOS* da DATA GENERAL permitem indexação hierárquica.**

Os métodos de acesso ao arquivo não estão cientes de qualquer estrutura que esteja abaixo do nível de registro: eles vêem um registro como uma cadeia de caracteres, algumas vezes com uma ou mais chaves. As linguagens de programação mais comuns que têm recursos para manipular dados agregados, como o COBOL e PL/1, selecionam campos nomeados individualmente com base numa declaração de estrutura no programa. Nesta declaração o programador define a seqüência exata dos campos e as características de cada um. No tempo de compilação qualquer referência aos campos individuais é traduzida em *off-sets* a partir do começo do registro pelo uso da descrição na declaração. Em outras palavras, a declaração fornece uma máscara que é colocada sobre o registro a fim de interpretá-lo.

Apesar de ser possível termos um arquivo que seja autodefinível, no sentido de que ele contém dentro dele informações para delimitar e descrever os campos, este método é raramente usado.

É muito difícil comparar o MUMPS com as técnicas tradicionais de manipulação de arquivos porque a terminc logia e os conceitos são muito diferentes. Entretanto, há muitos pontos de analogia e as construções análogas serão comparadas.

No MUMPS, a unidade básica de armazenamento é o "nodo". Um "nodo" de dados é uma cadeia de caracteres de tamanho variável com um máximo de 255 caracteres (nos dialetos anteriores os tamanhos máximos eram de 73, 95 ou 132). Um "nodo" corresponde muito a um registro, na medida em que ele é tanto a menor quanto a maior unidade que pode ser recuperada ou armazenada. O método de acesso usado é o de indexação hierárquica, com cada "nodo" definido por um ou mais "subscritos". Um "subscrito" é uma chave de tamanho variável, que pode somente conter dígitos. Nota do tradutor: esta restrição não existe no MUMPS da MEDIDATA, com árvore balanceada – neste caso, os subscritos podem ser alfanuméricos.

A linguagem não fornece recursos para manipulação automática de itens de dados ou campos dentro de um "nodo". Entretanto, ela tem poderosos recursos de processamento de cadeias de caracteres que permitem ao programador de aplicações simular os recursos que faltam. A capacidade de extrair uma seqüência de caracteres de uma cadeia ou construir uma cadeia que seja a concatenação de outras cadeias permite ao programador armazenar um item de dado como uma seqüência de caracteres. Todos os programas que se referem àquele item devem endereçar a mesma parte da cadeia. Esta técnica exige campos de tamanho fixo. Um método mais útil conta com a presença da função *piecing* (remendo), que extrai uma subcadeia delimitada por um caractere conhecido ou seqüência conhecida de caracteres.

Assim, um grupo de itens de dados de tamanho variável pode ser concatenado em uma cadeia única se estiver separado por algum caractere que nunca seja encontrado nos itens individuais. Um programa pode então referenciar-se ao primeiro, segundo, etc. . . item, numa cadeia de itens. Novamente este método exige que cada programa que manuseie um item de dados tenha consigo o conhecimento do delimitador usado e a posição relativa do item dentro da cadeia. Estas duas técnicas básicas algumas vezes são combinadas e por convenção pode haver delimitadores de maior e menor abrangência, de modo a permitir uma pequena estrutura hierárquica dentro de um único "nodo".

Naturalmente, os programadores podem decidir armazenar um único item de dado como um "nodo". Isto simplifica grandemente a programação e o processamento de caracteres. Entretanto, o grupamento de vários campos de dados dentro de um "nodo" é largamente empregado pelos programadores MUMPS, por várias razões importantes:

- Existe freqüentemente uma necessidade lógica de se agrupar os dados em uma única unidade.
- Combinando-se "n" itens em um "nodo" pode-se reduzir o número de acessos a disco para recuperação ou atualização de dados, por aproximadamente um fator "n", se os dados são geralmente necessários ao mesmo tempo.
- Alguma capacidade de armazenamento é poupada porque o que o sistema consome para cada "nodo" adicional, geralmente o equivalente a 5 caracteres, é mais do que os caracteres extra empregados como delimitadores.

ESTRUTURA ACIMA DO NÍVEL DE NODO (O REGISTRO)

O MUMPS vai muito além da gerência de dados tradicionais, permitindo que registros ("nodos") sejam interconectados hierarquicamente. Assim, um "nodo" pode ser o filho de um outro "nodo" e o pai de qualquer número de "nodos". As redes não são permitidas: cada "nodo" pode ter apenas um pai imediato, apesar da possibilidade de haver qualquer número de gerações na estrutura, e nenhum "nodo" pode ter um pai como descendente. É possível que haja "nodos" que sirvam somente como conectores, isto é, que não contenham dados. Estes "nodos" são freqüentemente chamados de "ponteiros", embora eles não devessem ser confundidos com uso tradicional desse termo. O conjunto de subscritos ou chaves que define um "nodo" é, na verdade, o caminho que vai do topo da hierarquia até o "nodo". Desta forma, há uma única chave associada à descendência (origem) de um pai para um filho. O MUMPS fornece os recursos para o endereçamento de um "nodo", para o armazenamento ou recuperação pela seqüência completa de chaves ou por uma parte da seqüência. Quando uma parte é usada, as chaves que faltam são fornecidas pelo sistema baseado na referência mais recente ao banco de dados. É possível também testar a existência ou não de um "nodo" arbitrário e se ele tem ou não algum dado. Existe também um meio de acessar seqüencialmente um conjunto de nós irmãos, isto é, dada uma posição arbitrária dentro da hierarquia encontra-se a próxima e única chave numericamente maior. Esta técnica não pode ser usada para passar de um conjunto de irmãos para outro. Em geral, há recursos para mover-se de pai para filho e de irmão para irmão; mas a única maneira de retornar é começar de novo do alto.

Este esquema de interconexão de "nodos" é muito útil porque muitos sistemas do mundo real podem ser repre-

NÚCLEO DE ENSINO
DE TECNOLOGIA E CIÊNCIA LTDA.

O NETC, com o compromisso de CONTRIBUIR para o aperfeiçoamento e a especialização profissional tecnológica, através de seu corpo Docente formado por Doutores e Mestres em Ciências e Engenheiros, oferece para o segundo semestre de 83 os seguintes cursos:

Fornecemos com seriedade e competência o KNOW-HOW que VOCÊ procura!

	CARGA HORÁRIA (Hs)
• Instrumentação e Controle I	60:00
• Conversão A/D e D/A na Instrumentação Eletrônica	35:00
• Sistemas de Comunicações por Fibras Ópticas	40:00
• Amplificadores Operacionais I e II	50:00 e 40:00
• Eletrônica Digital I - Combinacional	60:00
• Eletrônica Digital II - Sequencial	60:00
• Hardware de Teleprocessamento	45:00
• Software de Teleprocessamento	60:00
• Computadores e Microcomputadores Digitais-Hardware	45:00
• Memórias I - Monolíticas	35:00
• Hardware do Microprocessador 8080/85	35:00
• Hardware do Microprocessador Z-80	35:00
• Interfaceamento e periféricos do MP 8080/85	40:00
• Interfaceamento e periféricos do MP Z-80	40:00

Todos os cursos são INTEGRALMENTE Apostilados e Retroprojetados dispondo, em condições particulares, de Laboratório para AULAS PRATICAS, bem como: Filmes e Slides.
Também planejamos e executamos programas de formação profissional para Empresas, que permitem benefícios fiscais do CFMO (Lei 6.297/75).

Venha conhecer nossos Laboratórios, informar-se e inscrever-se à Rua Alvaro Alvim nº 37, 2º andar -Centro - RJ (Em frente à Estação Cinelândia-Metrô) de 2ª a 6ª feira das 16:00 hs às 22:00 hs. Sábados, das 8:00 hs às 17 Hs.

Traga outros candidatos e obtenha descontos - em quaisquer cursos!

ELETRODATA
INFORMÁTICA LTDA.

MICROCOMPUTADORES
• DGT-100/101 • CP200 • CP300 • CP-500 • TK-85 • ALFA 3000
EM ATÉ 24 MENSALIDADES.

AUTORIZADA:

PROLOGICA microcomputadores
dismac
DIGITUS
MICRODIGITAL

PROGRAMAS
• DE ENTRETENIMENTO
• PARA EMPRESA
• PARA MEDICINA
• DE ENGENHARIA
• DE ENSINO
• SOB ENCOMENDA

PERIFÉRICOS
• IMPRESSORAS
• L PRINT EM IBM
• DISK DRIVERS
• DISQUETES
• GRAVADORES ESPECIAIS COM MONITOR, INDICADOR DE NÍVEL E AJUSTE DE AZIMUTE

ASSISTÊNCIA TÉCNICA
TEL.: (021) 288-2650
RUA GENERAL ROCA, 675 G. 204
PÇA. SAENS PEÑA - TIJUCA, AO LADO DA CAIXA ECONÔMICA - RIO DE JANEIRO.

sentados por uma hierarquia ou até algo mais simples; por exemplo, uma hierarquia inclui uma organização indexada simples como subconjunto. Assim, uma entidade complexa pode ser representada como sendo composta de sub-entidades, cada uma das quais podendo também ser subdividida. Além disso, há freqüentemente a necessidade de estruturas em rede. Estas relações podem ser criadas pelo programador de aplicações. Por exemplo, apesar do MUMPS não fornecer arquivos invertidos, os programas MUMPS constroem tais arquivos pelo armazenamento de referências lógicas a outros "nodos". Por exemplo, em um sistema de controle de estoques o arquivo de peças pode ter o código da peça como chave. Um arquivo de fornecedores poderia apontar de forma lógica o arquivo de peças por meio dos códigos de peça.

Os modernos sistemas de gerenciamento de banco de dados estão preocupados, entre outras coisas, com a interconexão entre os registros. Há duas categorias principais de tais sistemas: hierárquica e de rede; até hoje bancos de dados relacionais não têm sido aplicados extensivamente nos sistemas comerciais de banco de dados.

O IMS da IBM é o sistema hierárquico mais amplamente usado. Ele inclui algumas capacidades limitadas de rede, que não faziam parte do sistema originalmente. Ele não é uma linguagem, exceto pelos conjuntos de comandos de descrição de banco de dados. Pelo contrário, ele é usado em conjunto tanto com o COBOL como com o PL/1 ou com a linguagem assembler do IBM 360/370, por meio de chamadas de sub-rotinas IMS. A unidade individual de armazenamento é chamada de *segmento*. Os segmentos podem ser de tamanho variável em algumas versões do IMS. Um segmento é, por sua vez, composto de campos. Um máximo de 255 tipos de segmentos diferentes são permitidos. Uma estrutura hierárquica deve ser pré-definida em termos dos tipos de segmentos que ela contém, embora o número de ocorrências de cada tipo de segmento possa variar dinamicamente durante o tempo de execução. Assim, o programa pode adicionar um segmento a uma hierarquia se, e na posição onde, tiver sido feita a previsão para aquele tipo de segmento na fase de descrição do banco de dados.

Em qualquer nível de hierarquia podem ocorrer múltiplos tipos de segmentos.

Como acontece em MUMPS, cada segmento pode ser referenciado pela seqüência completa de chaves que determinam seu caminho desde o topo da hierarquia. As chaves são cadeias de caracteres alfanuméricos que podem ser processados como dados pelos programas da linguagem hospedeira. Existem funções para encontrar o próximo elemento num dado nível sob o mesmo pai (como em MUMPS) e para encontrar o próximo segmento no mesmo nível que não diga respeito ao pai. É possível, como em MUMPS, referenciar um segmento por uma chave parcial baseada na referência anterior.

Um acréscimo especialmente forte do *get next* do IMS é a possibilidade de especificar argumentos de pesquisa condicionais. Assim, pode-se pedir o próximo segmento que tem um valor particular em um campo especificado. Condições complexas podem ser usadas, empregando-se combinações de operadores *relacionais* e *booleanos*.

Cada programa de aplicação que necessita manusear um dado tipo de segmento deve conter uma declaração que

Traduções técnicas
Inglês/Português . Português/Inglês
exclusivas
para área de informática

- Traduções exclusivas para a comunidade de informática.
- Pioneiros no setor
- Manuais e publicações diversas
- Gráficos
- Supervisão técnica de profissionais atuantes-analistas e consultores
- "Linguagens de alto nível" aliadas à qualidade/velocidade de entrega
- Informações pelos tels. (021) 264-6392/264-7391/228-2798

• • •

DISCOVER TRADUÇÕES

End.: Rua Professor Gabizo, 225 · Tijuca · Rio de Janeiro · RJ · Cep. 20271

defina uma área de trabalho para conter aquele segmento. Esta declaração inclui um catálogo dos campos dentro do segmento, e é o método pelo qual os campos são extraídos do segmento. É necessário que o sistema forneça a cada usuário área de trabalho suficiente para a residência simultânea de todos os tipos de segmento que ele tenha declarado.

Uma das diferenças mais marcantes entre o IMS e o MUMPS é o número de blocos de controle e linguagens de processamento exigidos pelo IMS. Enquanto o MUMPS manipula dinamicamente as estruturas de banco de dados com uma única linguagem, o IMS exige um complexo conjunto de blocos de controle em várias linguagens. Por exemplo, a linha seguinte em código MUMPS:

```
READ' NAME = , NAME, NUMBER = ' ,
NUM SET INDEX (NUM) = NAME
```

Poderia ser executada diretamente em qualquer máquina MUMPS, sem qualquer iniciação. A linha de código provoca a exibição de 'NAME=', aceita a resposta na variável NAME, exibe 'NUMBER = ' e aceita a resposta na variável NUM. Se o banco de dados INDEX não existir ainda, o MUMPS criará primeiro um banco de dados vazio com esse nome e depois colocará no "nodo" NUM o valor da variável NAME. Observe que nenhuma declaração de variáveis, tipos de dados ou tamanhos de cadeias é exigido; eles são manuseados pelo MUMPS dinamicamente.

Em contraste, para realizar a mesma função em IMS teríamos que definir pelo menos um DBD (Data-Base Descriptor), dois PSBS (Program Specification Blocks) um ACB (Accumulated Control Block), um MID (Message Input Descriptor), um MOD (Massage Out put Descriptor) e

um DIF/DOF (Device Input Format/Device Out put Format). O código de transação e o nome do PSB devem ser definidos em uma geração do IMS e num programa escrito, compilado e linkeditado a outros módulos do IMS. Os tamanhos dos campos (NAME e NUM) devem ser especificados pelo menos sete vezes, usando várias linguagens. ■

micro sem programa?
soft da monk nele.

Monk, soft pronto para ser usado.

PARA D.8000, CP 500, DGT 100 e NAJA

Cadastro, Banco de Dados, Locações, Contabilidade, Contas a Pagar e Receber, Editor de Texto, Conta Bancária, Mala Direta, Visicalc, Controle de Estoque, Editor Assembler, Compiladores Basic e Cobol, jogos que ninguém é de ferro. Relação com 60 programas, todos em disponibilidade agora. Todos em português, gravados em cassette ou diskette, com manual do usuário, extremamente práticos.
Procure no seu revendedor predileto nossa relação completa de programas, ele está em condições de aconselhá-lo e dar demonstrações técnicas.
Livre-se já dos custos em ORTN's e esperas duvidosas.
Sem soft seu micro não trabalha.

monk micro informática ltda.
R. Augusta, 2690 - 2.º andar - Loja 318
Tel. (011) 852-2958 - cep 01412 - SP

monk,
o software que faz você ficar feliz por ter um micro.

POLIEDRO

LIVRARIA POLIEDRO
Rua Aurora, 704 (Metrô República)
01209 — São Paulo, SP
tels.: 222-4297 — 221-6764

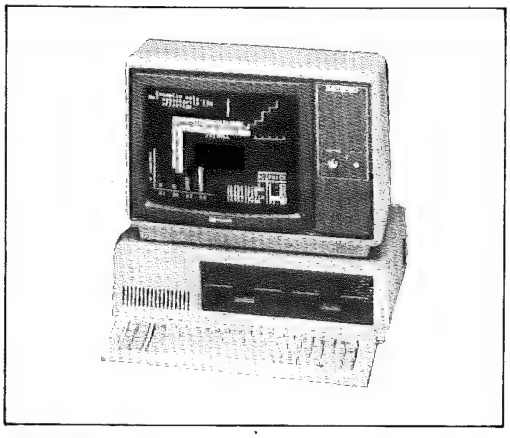
— BUILD YOUR OWN Z80 COMPUTER, Ciarcia . . .	9.000
— COMPUTES, GRILO!, Pereira	2.600
— APLICAÇÕES SÉRIAS (TK, NEZ CP) Lima	3.000
— 45 PROGRAMAS PRONTOS PARA RODAR (TK, NEZ, CP) Lima	3.000
— INTRODUÇÃO AO VISICALC, Garbin	2.300
— BASIC PARA MICROS PESSOAIS, Pereira	2.950
— CRIANÇA TAMBÉM FAZ PROGRAMAS, Silva . . .	1.390
— MICROPROCESSADORES 8080/8085 — Hardware, Visconti	2.800
— MICROPROCESSADORES 8080/8085 — Software, Visconti	3.200
— DICIONÁRIO DE INFORMÁTICA, Sucesu	8.900
— DICIONÁRIO DE TERMOS CHAVES DE COMPUTADORES, Bussoti	800
— CP/M — Guia do Usuário, Osborne	a sair
— INTRODUÇÃO AOS MICROCOMPUTADORES, vol. O, Osborne	a sair
— DIGITAL SYSTEMS: Hardware Organization/Design, Hill	15.000
— 30 JOGOS PARA TK NEZ CP — Lima	2.500

Pedidos pelo reembolso POSTAL ou VARIG para Caixa Postal 386 — 01000 — São Paulo — SP

OUTUBRO/83

Interface

mercado de computadores



Como escolher um computador que seja adequado às suas necessidades? Hoje os preços de computadores já são acessíveis, e todos já podem possuir um, inclusive você.

Selecione aqui o seu microcomputador ou periférico e saiba onde adquiri-lo nos mercados do Rio, São Paulo, Belo Horizonte etc.

Características Técnicas & Preços

AO SER HUMANO CABE CRIAR, À MÁQUINA PRODUZIR.

Desde a criação da Terra o homem saiu na frente dominando-a, graças à sua capacidade de criar. E, uma das maiores violações impostas a ele são os trabalhos repetitivos onde se proibe o "Estado da Arte", onde todo o tempo a tarefa é produzir, repetindo movimentos homogêneos horas após horas.

Pensando nisto criou-se a *ENAC*, empresa de Projetos Eletrônicos voltada a Periféricos e Sistemas Dedicados, dedicando-se a projetos que atendam o mercado carente de soluções práticas por custo baixo, viável na atual conjuntura sócio-econômica. Dentro de sua linha de produção lança ao mercado consumidor o produto *LPRINT*.

Imagine o trabalho de uma secretária datilógrafa, o tempo que ela perde redatilografando por causa de erros ou alterações de textos originais, além dos trabalhos repetitivos como: procurações, recibos, contratos, orçamentos, etc. Todos estão de acordo que estas atividades representam um tempo perdido no que se refere ao aperfeiçoamento da capacidade de criar que todos nós possuímos. A LPRINT possibilita que através de um computador estes trabalhos repetitivos não existam, deixando por conta da máquina esta atividade desgastante e antiprodutiva.

LPRINT uma interface que permite ligar qualquer computador a uma simples máquina de escrever elétrica (tipo IBM de esfera). Uma solução simples, barata e eficiente, sendo suas qualidades inúmeras tais como:

- Qualidade de escrita incomparável.
- Caracteres como cedilha e acentuação da Língua Portuguesa.
- Geração de negritos e grifos
- Aceita qualquer tipo de papel (formulário contínuo, envelopes, papel timbrado, etc.).
- Permite uma interação impressão pelo computador e pelo teclado da máquina.
- A máquina IBM continua com suas características originais.
- 16 CPS (velocidade máxima da máquina IBM).
- Facilidade de Assistência Técnica.

A ENAC atua ainda no setor de Sistemas Dedicados, desenvolvendo microcomputadores para controle e automatização industriais, além de implantação e suporte a instalações de sistemas na área de informática, fornecendo software e treinamento.

A ENAC tem sua sede em Campinas à Rua Coronel Quirino, 501 — fone (0192) 52-0964 — Caixa Postal 1865 — CEP 13.100 e representação em São Paulo através da Microprint — Av. Moreira Guimarães, 538 — fone (011) 61-8796 — CEP 04074.

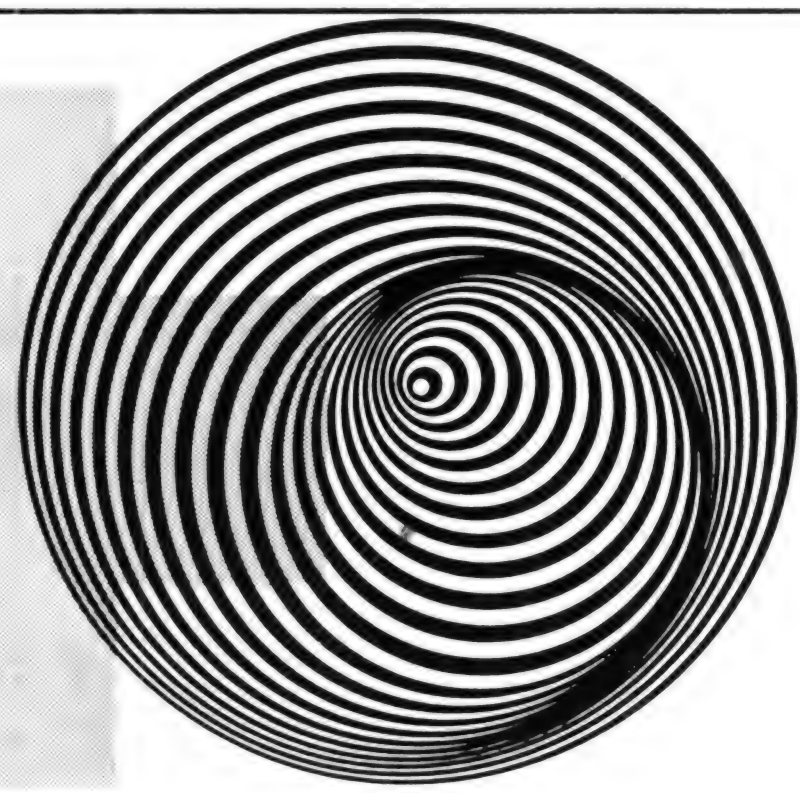
Micro Consult

CONSULTORIA DE SISTEMAS EMPRESARIAIS
E MICROPROCESSAMENTO LTDA

Sistema modular para gerenciamento de Clínicas médicas utilizando microcomputadores, consistindo dos seguintes módulos:

- Agenda Médica
- Arquivo Médico
- Correspondência
- Controle de estoque
- Programa de desembolso
- Programa financeiro
- Faturamento
- Laudas
- Sistemas específicos

Av. N. S. de Copacabana, 605 / 1210
Tel. (021) 236-1325
Telex. (021) 32479 AUDO
RIO DE JANEIRO



DIAS 16 E 17 DE SETEMBRO DE 1983

1º SEMINÁRIO SOBRE O MICROCOMPUTADOR NA ÁREA MÉDICA.

Apoio:

SOCIEDADE BRASILEIRA DE MEDICINA E CIRURGIA - RJ.

O I.º Seminário sobre Aplicações do Microcomputador na Clínica Médica tem o objetivo de transmitir ao Médico, estudante de Medicina e outros profissionais ligados a administração Hospitalar, as múltiplas aplicações do microcomputador, conceitos básicos de hardware e software, características e potencialidades do microcomputador, aplicações mais comuns e específicas, e onde melhor utilizá-lo no consultório, hospital e etc.

Para atingir esses objetivos foram convidados profissionais de informática de renome e profissionais da área médica altamente qualificados que desenvolvem trabalhos na área de microcomputação.

Local: COLÉGIO BRASILEIRO DE CIRURGIÕES.
R. Visconde Silva, 52 - Botafogo - RJ.

Informações e Inscrições

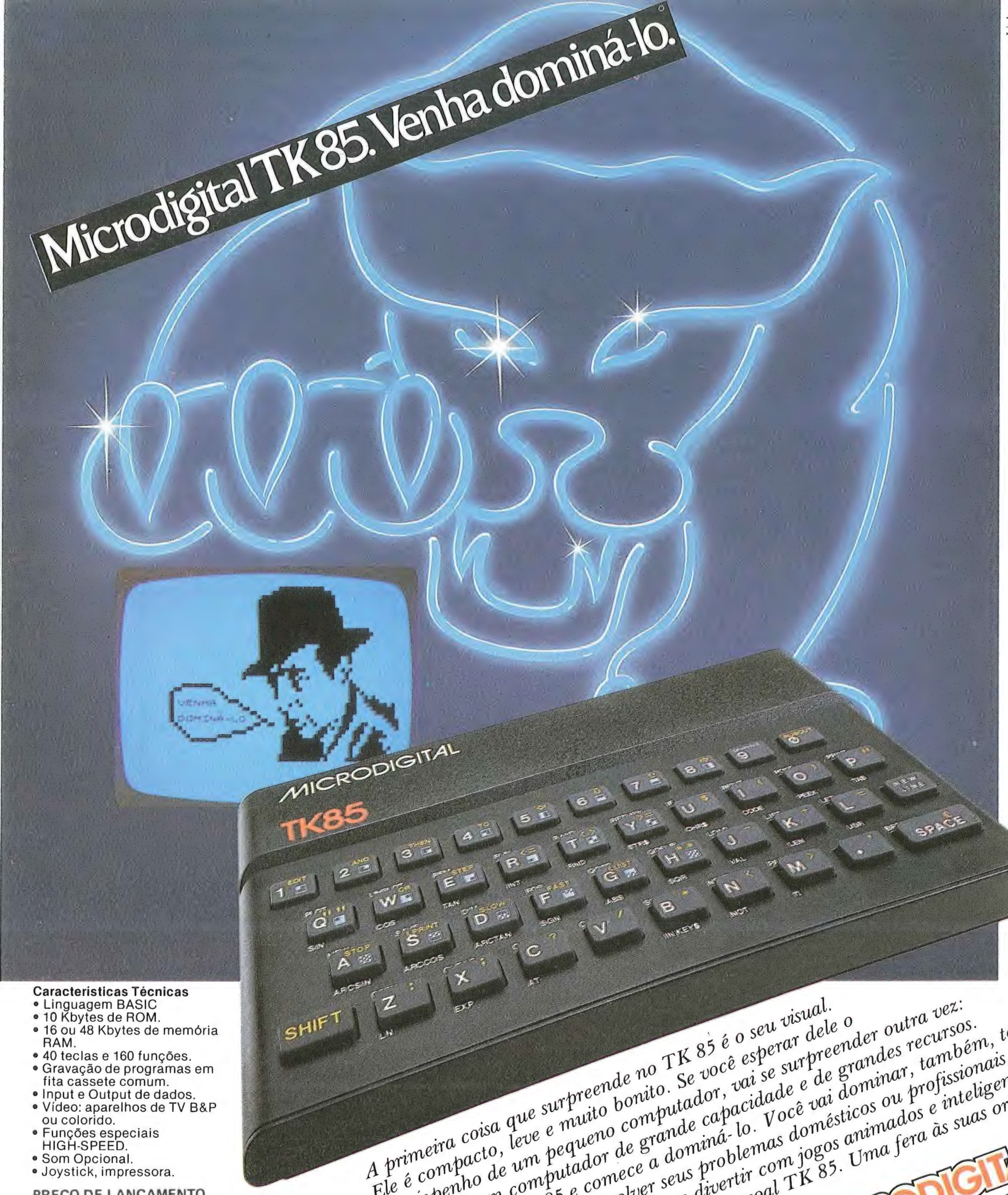
Sociedade de Medicina e Cirurgia do Rio de Janeiro.
Av. Mem de Sá, 197 - Tel.: 232-2888

CBI - Centro Bras. de Informática.
Av. Passos, 115 - sala 215
Tel.: 233-1123

COLABORAÇÃO **ROCHE**

Patrocínio Interface REVISTA

Microdigital TK 85. Venha dominá-lo.®



Características Técnicas
- Linguagem BASIC
- 10 Kbytes de ROM.
- 16 ou 48 Kbytes de memória RAM.
- 40 teclas e 160 funções.
- Gravação de programas em fita cassete comum.
- Input e Output de dados.
- Vídeo: aparelhos de TV B&P ou colorido.
- Funções especiais HIGH-SPEED.
- Som Opcional.
- Joystick, impressora.

PREÇO DE LANÇAMENTO
Cr$ 198.850,00 (16K)
Cr$ 269.850,00 (48K)
(Preço sujeito a alteraçao)

A primeira coisa que surpreende no TK 85 é o seu visual. Ele é compacto, leve e muito bonito. Se você esperar dele o desempenho de um pequeno computador, vai se surpreender outra vez: o TK 85 é um computador de grande capacidade e de grandes recursos. Acione o TK 85 e comece a dominá-lo. Você vai dominar, também, todas as situações. Resolver seus problemas domésticos ou profissionais. Vencer desafios e se divertir com jogos animados e inteligentes. Computador Pessoal TK 85. Uma fera às suas ordens.

MICRODIGITAL
Rua do Bosque, 1.234 - Barra Funda
CEP 01136 - Cx.P. 54.088 - São Paulo - SP
PABX 825-3355

REVENDEDORES: ARACAJÚ 224-1310 • **BELEM** 222-5122/226-0518 • **BELO HORIZONTE** 226-6336/225-3305/225-0644/201-7555 • **BLUMENAU** 22-1250 • **BRASÍLIA** 224-2777/225-4534/226-9201/226-4327/242-6344/242-5159 • **BRUSQUE** 55-0675 • **CAMPINAS** 32-3810/8-0822/32-4155/2-9930 • **CAMPO GRANDE** 383-6487/382-5332 • **CARUARU** 721-1273 • **CUIABÁ** 321-8119/321-7929 • **CURITIBA** 232-1750/224-6467/224-3422/243-1731/223-6944/233-8572/232-1196 • **DIVINÓPOLIS** 221-2942 • **FLORIANÓPOLIS** 23-1039 • **FORTALEZA** 226-4922/231-5249/231-0577/231-7013 • **FREDERICO WESTPHALEN** 344-1550 • **GOIÂNIA** 261-0333/224-0557 • **IJUÍ** 332-2740 • **ITAJUBÁ** 622-2088 • **LINS** 22-2428 • **LONDRINA** 22-4244/23-9674 • **MACEIÓ** 223-3979/221-6776 • **MANAUS** 237-1793 • **MOGI DAS CRUZES** 468-3779/208-6797 • **MURIAÉ** 721-1593 • **NATAL** 222-3212/231-1055 • **NITERÓI** 722-6791 • **NOVO HAMBURGO** 93-1922/93-3800 • **PELOTAS** 24-5139 • **PORTO ALEGRE** 26-8246/21-4189/24-1411/22-3151/24-0311/21-6109/24-7746 • **PRESIDENTE PRUDENTE** 22-2788 • **RECIFE** 241-4310/224-8777/224-3436/224-4327 • **RESENDE** 54-1664 • **RIBEIRÃO PRETO** 636-0586/634-4715/635-1195 • **RIO DE JANEIRO** 267-1093/252-2050/253-3395/264-0143/259-1516/232-5948/591-3297/222-6088/267-1339/329-4869/228-2650/246-4824/239-5612/542-3849/62-8737 • **SALVADOR** 248-6666/235-4184/247-5717 • **SANTA MARIA** 221-7120 • **SANTO ANDRÉ** 455-4962/444-7375/454-9283 • **SANTOS** 4-1220/32-7045/35-1792/33-2230 • **SÃO CARLOS** 71-9424 • **SÃO JOÃO DA BOA VISTA** 22-3336 • **SÃO JOSÉ DOS CAMPOS** 22-3968/22-7311/22-8925/21-3135 • **SÃO PAULO** 853-0164/853-0448/239-4122/36-6961/61-4049/881-1149/258-3954/212-9004/282-2105/212-3888/545-4769/227-3022/864-8200/222-1511/259-2600/282-6609/813-4555/814-3663/826-1499/521-3779/270-7442/210-7681/813-4031 • **SOROCABA** 32-9988 • **TAUBATÉ** 31-4137 • **UBERABA** 333-1091 • **UBERLÂNDIA** 234-8796 • **VIÇOSA** 891-1790/891-2258 • **MARÍLIA** 33-4109

Link

balcão

CLASSIFICADOS DO LEITOR: REMETA SEU ANÚNCIO PARA REVISTA INTERFACE – MERCADO DE COMPUTADORES. ESTRADA DO TINDIBA, 2380. JACAREPAGUÁ – CEP 22700 – RIO DE JANEIRO – RJ.

SOFTWARE CP/TK/NE – Vendo 90 programas importados diversos para CP-200 e TK; jogos como King-Kong do fliperama, aplicativos etc.
Tratar pelo telefone 449-3424 com Júnior ou na Av. Senador Vergueiro, 2685 Bloco 11-A apto. 104 ou 132. Rudge Ramos – São Bernardo do Campo.

Vendo programas de origem européia para TK82, CP200 e similares, gravados em fita K7. Tratar com Alexandre – Rua Pedro, 801 – Tremembé – CEP 02371 – São Paulo – SP ou pelo telefone 203-4277.

GRANDE CIRCUITO – Anúncios grátis. Compras, vendas, trocas. Para usuários de TK/CP-200/Zx etc. Informações pela Caixa Postal 28 – CEP 27200 – Piraí – RJ.

2K serial interface buffer board for MX-100. Parallel interface board/cable for aplle II/MX-80, ou troco por eprom programmer. Tratar pelo telefone 263-1880, com Flavio.

FAÇO MODIFICAÇÕES: Slow, controle de gravador funções especiais, gravador de Eprom nos micros
CP200. Também gravo memórias sobre encomendas. Tratar com Genilton de Castro Brito – Rua 227-A nº 255 – Setor Universitário – Goiânia – Tel.: 261-2862.

Estabilizador de tensão BK, 0.8 KVA, 220/220 volts. Novo. Tratar na rua Flack, 135 C/1 – Rio de Janeiro – CEP 20960 – Telefone 261-8783/261-7018. Troco Printer.

Vende-se fitas de jogos animados para TK-82C, TK-85, CP-200 e ZX-81 (TK-Man, Simulador de Vôo, Monstro das Trevas, Mazogs e muitos outros). Tratar com Michel pelo telefone (011) 288-8878 ou com Isidoro no (011) 853-9893.

Preciso de manual ou xerox do manual do **Disk-Basic** e do DOS do Newdos-80, versão 2.0 para o TRS-80 modelo I. Quem tiver, por favor entrar em contato com Beth (711-7844) ou Eraldo (719-9265) à noite.

Calculadora HP-41C. Vende-se 180 mil, tratar com Arlindo Martins Filho na Rua Esmeraldino Bandeira, 393 – Aflitos – Recife-PE – CEP 50000 (061) Telefone 222-4925.

Vendo TK82-C com expansão de 16K RAM, novo (com três meses de uso), c/SLOW/FAST. Vendo para aquisição de um de maior porte, por Cr$ 100 mil, acompanha dois jogos. Tratar com José T. Tavares Jr. na Av. Vicente de Carvalho, 92/124 – Santos – SP – CEP 11100.

Vendo programa de banco de dados com informações precisas e atualizadas sobre **ANTIBIOTICOTERAPIA.** Útil para médicos que possuam TK/NE/Sinclair com 16K e queiram utilizá-lo no consultório. Também disponho de outros bancos de dados médicos. Preço Cr$ 18.000,00. Solicite maiores informações ao Dr. Lucio A. Castagno na rua General Osório, 1585 – Pelotas-RS – CEP 96100.

Vendemos um TK-82 junto com a expansão (16-K). O equipamento está novo e a expansão ainda tem garantia até final de julho; estamos pedindo Cr$ 140.000,00 (cento e quarenta mil cruzeiros). O motivo da venda é porque estamos interessados em adquirir um TK-85 ou um CP-200. Para maiores esclarecimentos ligar para 224-9788 (horário comercial) e falar com Annette.

LIVROS PARA TK, NEZ, CP

45 PROGRAMAS

Prontos para Rodar

ARQUIVOS, ESTOQUE, PLANO CONTÁBIL, AGENDA TELEFÔNICA, INVASORES, CAÇA AO PATO, APAGUE A TRILHA, JOGO DA VELHA, FORCA, DADO, TABELAS, TABUADAS, CONVERSÃO DE COORDENADAS, MÉDIA, PROGRESSÃO, FIBONACCI, BIO RITMO, RENUMERADOR DE LINHAS EM CÓDIGO, ETC... ETC...

5 Edicao **CR$ 3.000,00**

APLICAÇÕES SÉRIAS

C/ Programas Listados por Impressora.

FOLHA DE PAGAMENTO, BALANCETE, CONTAS A RECEBER, A PAGAR, CORREÇÃO MONETÁRIA DAS CONTAS DO BALANÇO, CORREÇÃO DAS CONTRIBUIÇÕES DO IAPAS, CADASTRO DE CLIENTES, CONTA BANCÁRIA, TABELA PRICE, ESTATÍSTICA, CORREÇÃO DE PROVAS, EDITOR DE TEXTOS, RAM TOPER. SUBROTINAS EM CASSETTE, CHAINING PROGRAMAS, CONTANDO OS BYTES DAS LINHAS, DO PROGRAMA, DAS MATRIZES, ECONOMIZANDO MEMÓRIA, ETC... ETC...
INCLUINDO:
CONHECENDO A IMPRESSORA, VALE A PENA? VEJA AMOSTRA DO PAPEL.
PROJETO COMPLETO DE TECLADO MECÂNICO, COM LAY-OUT DOS CIRCUITOS IMPRESSOS, DOS PAINÉIS E GABINETE, ETC...

CR$ 3.000,00

FIRMWARE PARA TK, NE Z

Circuito Leitor de Eprom **CR$17.500,00**

Ligado entre a expansão de memória e o micro, lê programas de uma EPROM pré-programada, com o uso da função USR. É só ligar para usar os programas. Não requer alimentação extema ou alterações no micro.

Eprom's Pré-Programadas

Funções I	Cr$ 9.500,00
Ram Toper	Cr$ 9.500,00
Paredão (jogo)	Cr$ 9.500,00

SOFTWARE EM CASSETTE

Solicite relação de programas.

À VENDA NAS LOJAS ESPECIALIZADAS
DESPACHAMOS PARA TODO O BRASIL MEDIANTE CHEQUE NOMINAL COM 10% PARA FRETE E EMBALAGEM.

MICRON
ELETRÔNICA COMÉRCIO E INDÚSTRIA LTDA.
Av. S. João 74 - Telefone 22-4194 S. José dos Campos Est. de São Paulo

TRANSITRON

Comércio de Componentes

- TTL
- C.MOS
- ICL 7107
- 2114
- 2708
- 2716
- 2732
- LINHA Z80
- LINHA Z80A
- ELETROLÍTICO
- TRANSÍSTOR
- POLIESTER
- TÂNTALO
- PLATE
- RESISTOR
- FUSÍVEL
- SOQUETE
- CONECTOR C. IMP.
- APAGADOR DE EPROM

Cr$ 55.000,00

TRANSITRON ELETRÔNICA LTDA.
Rua dos Gusmões, 353, 3º andar, cj. 31
Fones 221-2959 - 221-2701 - 223-5187
Telex - 37982 Representante em B. H.
Rua Eng. Antonio Guerra, 174, cj. 401
Fone 332-9580 - Sr. Rogério.

MC INFORMÁTICA

TUDO
EM MICROCOMPUTADORES

- Cursos de programação com APOSTILA PRÓPRIA E AULAS PRÁTICAS em diversos MICROCOMPUTADORES
- Todas as principais marcas de MICROCOMPUTADORES pelo menor preço com crédito direto em até 24 MESES.
- Programas prontos ou por encomendas tanto de jogos quanto comerciais.

VENHA CONHECER HOJE
A ERA DO FUTURO

MICROCENTER INFORMÁTICA LTDA.

Rua Conde de Bonfim, 229 – Lojas 310 e 312 – Galeria Cinema III
Tel. 264-0143 – CEP 20520 – Tijuca
– Rio de Janeiro - RJ

LOJA MICRO-KIT
TUDO SOBRE MICROCOMPUTADOR

FAÇA AGORA A SUA ENCOMENDA

LANÇAMENTO DO LIVRO "CURSO DE BASIC" – VOL. I – EDIÇÃO PRÓPRIA

CURSOS

Basic p/adultos e crianças, com método próprio comprovadamente eficiente. Professores c/mestrado em ENGENHARIA DE SISTEMAS, mais de 20 cursos aplicados. Turmas pequenas, aulas práticas com MICROCOMPUTADOR.

VENDA DE MICROCOMPUTADOR

Unitron AP II, Digitus, TK e CP 200.
Financiamento em até 24 meses.

PROGRAMAS

Comerciais e Jogos p/APPLE, Unitron, Polymax, Digitus TK e CP 200.

SUPRIMENTOS

Disquetes, Caixa p/Disquetes, Formulários Contínuos etc.

VENDA DE LIVROS E REVISTAS

Despachamos para todo o Brasil.

Rua Visconde de Pirajá, 303 S/Loja 210
Tels. (021) 267-8291 521-4638
CEP 22410 – Rio de Janeiro
Rua Visconde de Pirajá, 365 – Sobreloja 209
Ipanema

LITEC
Livraria Editora Técnica Ltda.

A maior livraria da América Latina especializada em INFORMÁTICA, COMPUTAÇÃO E ELETRÔNICA.
Mais de 3.000 títulos em português, espanhol e inglês em permanente exposição.

Rua dos Timbiras 257 - 01208 São Paulo Tel. (011) 220-8983 cx. postal 30.869

GDM INFORMÁTICA LIMITADA

- *Programas para TK ou CP Pronto em Fitas*
- *Curso de Basic*
- *Joystick (qualquer micro)*
- *Vídeo Reverso para TK*
- *Kit Treinador Lógico (Para est. Sist. Eletrônicos)*

MICROCOMPUTADORES
JR DA SISDATA
TK-82C - TK85
DIGITUS

Leasing e crédito direto

Assistência técnica especializada
(UNITRON AP-II, MICROENGENHO EXPANSÕES, IMPRESSORAS ETC.)

NA COMPRA DE QUALQUER MICRO (50% CURSO DE BASIC) E FITAS COM JOGOS

Rua Conde de Bonfim, 346 - Lj. 307
Ed. Vitrine da Tijuca
Pça S. Peña – Tel.: 284-8744

BITS & BYTES
COMPUTADORES

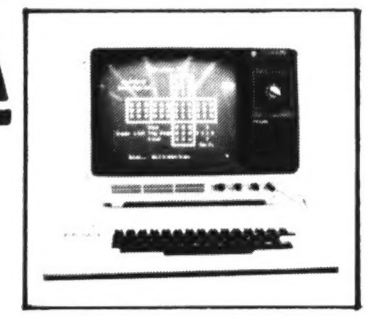
- VENDAS
- ASS. TÉCNICA ESPECIALIZADA
- PROGRAMAS
- DISKETTES
- FITAS
- SERVIÇOS
- CURSOS DE BASIC
- FORMULÁRIOS

CONSERTOS EM 24 HORAS (COM GARANTIA) PARA O CP-500 e DGT-100

EM SÃO CONRADO
Estrada da Gávea, 642 – Lj. B
TEL.: 322-1960

INSCREVA-SE JÁ

INFORMÁTICA



INFORMÁTICA

A INFORMÁTICA é a indústria que mais cresce no mundo
a que paga os melhores salários
e a única carente de técnicos especializados

FORMAÇÃO DE TÉCNICOS ESPECIALIZADOS

PROGRAMAÇÃO E MANUTENÇÃO DE MICROCOMPUTADORES

PRÉ-REQUISITO : 2º GRAU COMPLETO
DURAÇÃO : 920 HORAS
HORÁRIO : 19.00 às 22.30 HORAS
2ª a 6ª feira

Estágio Supervisionado
Certificado de Estabelecimento do Sistema Oficial de Ensino.

CURSOS EXTRA-CURRICULARES

- PROGRAMAÇÃO BASIC
- BASIC AVANÇADO
- LÓGICA DIGITAL (Grátis: Treinador Lógico)
- MICROPROCESSADORES – 8080/85 – Z-80
- SISTEMA OPERACIONAL CP/M
- COBOL

Visite nossos laboratórios

CENTRO EDUCACIONAL CARVALHO DE MENDONÇA

ENGEMICRO
ENGENHARIA EDUCACIONAL DE MICROPROCESSAMENTO

Rua Evaristo da Veiga, 20 - Tel. 220-8820/220-7009

CERTIFICADO: REGISTRO NO SISTEMA OFICIAL DE ENSINO

OFERTAS

Kristian

MICROCOMPUTADORES

DGT-100 Cr$ 220.000, x 3 — Grátis 18 JOGOS
CP-500 Cr$ 790.000, — Grátis 18 JOGOS
CP-200 Cr$ 100.000, x 2 — Grátis 18 JOGOS
TK82-C Cr$ 49.925, x 2 — Grátis 10 JOGOS
ainda UNITRON Ap II, Mem 64K, Impressoras, etc...
(Preços sujeitos a modificações)

PROGRAMAS PRONTOS EM FITAS

JOGOS (TK ou CP)

- VISITA AO CASSINO
- MIDWAY
- PASSAGEM PARA O INFINITO
- 10 JOGOS EXCITANTES PARA 1K
- 2ª DIMENSÃO
- JORNADA NAS ESTRELAS
- E MUITO MAIS!

JOGOS: (DGT-100 OU CP-500)

- SCARFMAN
- PENETRATOR
- SUPER NOVA
- VIAGEM A VALKYRIA
- ASILO 1
- AVENTURAS
- DEFENSE COMMAND
- E MUITO MAIS!

LEASING E CRÉDITO DIRETO!

LITERATURA

- MICRO-SISTEMAS
- INTERFACE
- JORNAL TK-CP
- IMPORTADOS

+ CURSOS DE BASIC GRÁTIS
NA COMPRA DE QUALQUER MICRO

DESPACHAMOS PARA TODO O BRASIL!

APLICATIVOS

- CONTROLE DE ESTOQUE
- CONTAS A PAGAR/RECEBER
- MALA DIRETA/CADASTRO
- FOLHA DE PAGAMENTO
- VIDEO-CLUBES
- ESTATISTICOS
- SOFTWARE SOB ENCOMENDA

Kristian
ELETRÔNICA LTDA.
Rua da Lapa, 120 Gr. 505
Rio de Janeiro - RJ
Tel.: (021) 252-9057

A CERTEZA DE UM BOM NEGÓCIO

TECCOR ELECTRONICS, INC

TIPO	CORRENTE DIRETA IF (RMS) A (max)	TENSÃO REVERSA VRRM V (min)	ENCAPSULAMENTO
RETIFICADORES CONTROLADOS DE SILÍCIO (SCR)			
S2001MS2	1,6	200	T05
S6003L	3	600	T0220AB
T106D1	4	400	T0202AB
T107D1	4	400	T0202AB
S2006L	6	200	T0220AB
S4015L	15	400	T0220AB
RETIFICADORES CONTROLADOS DE SILÍCIO BIDIRECIONAIS			
Q200E3	0,8	200 a 400	T092
Q2004F31	4	200 a 400	T0202AB
Q2008L4	8	200 a 600	T0220AB
Q2010L4	10	200 a 600	T0220AB
Q2015L5	15	200 a 600	T0220AB
TRIAC LÓGICO			
L4004F51	4	400	T0202AB
QUADRACS (TRIACS COM DIACS INCORPORADOS)			
Q4004FT1	4	400	T0202AB
Q4006LT	6	400	T0220AB
Q2008FT1	8	200	T0202AB

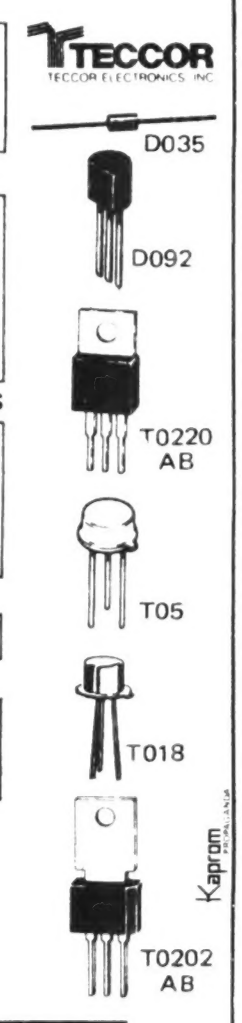
(TRIAC)

SCR ATIVADO POR LUZ

PR 30 — Tensão Bloqueio Direta Reversa 30 volts (min)
Corrente Bloqueio Direta Reversa 25 uA (max) (T018)

DISPARADOR BILATERAL DE SILÍCIO (DIAC)

HT32 — Vbo = 27/32t/37, 1,5A para pulso de 10uS de duração (D035)



Teleimport / Eletrônica Ltda.
Rua Sta. Ifigênia, 402, 8/10º andar - CEP 01207 - São Paulo
Fone: 222-2122 - Telex (011) 24888 TLIM-BR
(Solicite nosso catálogo geral de componentes)

CARACTERÍSTICAS TÉCNICAS

UNIDADE CENTRAL:

- 02 microprocessadores Z80 e 6502
- Memória RAM de 64KB (250NS), expandível p/128KB
- Monitor resistente em EPROM de 2KB
- Interpretador Basic residente em EPROM de 10KB
- Interfaces para:
  - Monitor de vídeo preto/branco
  - Monitor de vídeo a cores padrão PAL-M e RGB analógico
  - Gravador cassete de audio
  - Impressora paralela
  - MODEMS e demais periféricos seriais
  - Light-pen
  - Joystick
- Teclado profissional destacável
- Alimentação através de fonte chaveada de alto rendimento
- 06 (seis) conectores adicionais p/expansão compatíveis com Apple II*

PERIFÉRICOS:

- Controladores p/ até 06 (seis) unidades de disco flexível
- Unidade de disco flexível de 5 1/4 (128 KB), face simples e densidade simples
- Impressora paralela (Interface Centronics)
- Impressora Serial (RS 232-C)
- Monitor de vídeo (fósforo verde) de 24 linhas por 40/80 colunas

SOFTWARE:

- MX DOS
  - Compatível com DOS 3.3 do Apple II*
- Magnos
  - Compatível com CP/M*

OPCIONAIS PARA EXPANSÃO

- 128 KB de memória em RAM
- 2 (duas) unidades de disco flexível 5 1/4" (143 KB cada)
- 2 (duas) unidades de disco flexível de 8", formato IBM 3740 e também dupla face e dupla densidade capacidade 1 MB cada.

APLICAÇÕES:

- Nas empresas de pequeno porte, e no processamento distribuído nas empresas de médio/grande porte em aplicações nas áreas administrativa, financeira, comercial, estatísticas, etc.
- Na área Técnica-Científica, em aplicações específicas de pesquisa e desenvolvimento, permitindo tráfigos a cores em alta resolução (280h x 192v – em 8 cores).
- Para Profissionais Liberais, como computador pessoal

*Apple II e CP/M são marcas registradas de Apple Comp. Corp. e Digital Research Corp. – USA.

MAGNEX ELETRÔNICA LTDA.
Rua Dr. Thyrso Martins, 100
Vila Mariana – São Paulo
Tel: 570.2872 / 549.2232
Telex: (011) 4837 DLTD
CEP 04120

MG 8065 O Microcomputador que coloca o futuro ao seu alcance

MAGNEX

NOVO CP 300 PROLÓGICA.

O pequeno grande micro.

Compatível com programas em fita cassete ou em disco.

Agora, na hora de escolher entre um microcomputador pessoal simples, de fácil manejo e um sofisticado microcomputador profissional, você pode ficar com os dois.

Porque chegou o novo CP 300 Prológica.

O novo CP 300 tem preço de microcomputador pequeno. Mas memória de microcomputador grande.

Ele já nasceu com 64 kbytes de memória interna com possibilidade de expansão de memória externa para até quase 1 megabyte.

Pode ser acoplado a uma impressora.

E tem um teclado profissional, que dá ao CP 300 uma versatilidade incrível. Ele pode ser utilizado com programas de fita cassete, da mesma maneira que com programas em disco.

64K

O único na sua faixa que já nasce com 64 kbytes de memória.

Permite conexão telefônica.

Pode ser ligado ao seu aparelho de TV, da mesma forma que no terminal de vídeo de uma grande empresa.

Com o CP 300 você pode fazer conexões telefônicas para coleta de dados, se utilizar de uma impressora e ainda dispor de todos os programas existentes para o CP 500 ou o TRS-80 americano. E o que é melhor: você estará apto a operar qualquer outro sistema de microcomputador.

Pode ser ligado a um televisor comum ou a um sofisticado terminal de vídeo.

Nenhum outro microcomputador pessoal na sua faixa tem tantas possibilidades de expansão ou desempenho igual.

CP 300 Prológica.

Os outros não fazem o que ele faz, pelo preço que ele cobra.

PROLOGICA
microcomputadores

Av. Eng.º Luis Carlos Berrini, 1168 - SP

RESERVA DE MERCADO

DEFESA DOS VALORES NACIONAIS

AM
Manaus - 234-1045
• BA-Salvador - 247-8951
• CE-Fortaleza - 226-0871 - 244-2448
• DF-Brasília - 226-1523 - 225-4534 • ES-Vila Velha
229-1387 - Vitória - 222-5811 • GO-Goiânia - 224-7098 • MT
Cuiabá - 321-2307 • MS-Campo Grande - 383-1270 - Dourados - 421-1052
• MG-Belo Horizonte - 227-0881 - Betim - 531-3806 - Cel. Fabriciano - 841-3400 - Juiz
de Fora - 212-9075 - Uberlândia - 235-1099 • PA-Belém - 228-0011 • PR-Cascavel - 23-1538 - Curi-
tiba - 224-5616 - 224-3422 - Foz do Iguaçu - 73-3734 - Londrina - 23-0065 • PE-Recife - 221-0142 • PI-Teresina
222-0186 • RJ-Campos - 22-3714 - Rio de Janeiro - 264-5797 - 253-3395 - 252-2050 • RN-Natal - 222-3212 • RS-Caxias do
Sul - 221-3516 - Pelotas - 22-9918 - Porto Alegre - 22-4800 - 24-0311 - Santa Rosa - 512-1399 • RO-Porto Velho - 221-2656 • SP
Barretos - 22-6411 - Campinas - 2-4483 - Jundiaí - 434-0222 - Marília - 33-5099 - Mogi das Cruzes - 469-6640 - Piracicaba - 33-1470 - Ribeirão
Preto - 625-5926 - 635-1195 - São Joaquim da Barra - 728-2472 - São José dos Campos - 22-7311 - 22-4740 - São José do Rio Preto - 32-2842 - Santos - 33-2230
Sorocaba - 33-7794 • SC-Blumenau - 22-6277 - Chapecó - 22-0001 - Criciúma - 33-2604 - Florianópolis - 22-9622 - Joinvile - 33-7520 • SE-Aracajú - 224-1310.

Solicite demonstração nos principais magazines.